

Open Research Online

The Open University's repository of research publications and other research outputs

Negotiated Tutoring: An Approach to Interaction in Intelligent Tutoring Systems

Thesis

How to cite:

Baker, Michael J. (1990). Negotiated Tutoring: An Approach to Interaction in Intelligent Tutoring Systems. PhD thesis The Open University.

For guidance on citations see [FAQs](#).

© 1990 The Author



<https://creativecommons.org/licenses/by-nc-nd/4.0/>

Version: Version of Record

Link(s) to article on publisher's website:

<http://dx.doi.org/doi:10.21954/ou.ro.0000d386>

Copyright and Moral Rights for the articles on this site are retained by the individual authors and/or other copyright owners. For more information on Open Research Online's data [policy](#) on reuse of materials please consult the policies page.

oro.open.ac.uk

5 937
DX 92034
UNRESTRICTED

Negotiated Tutoring

An Approach to Interaction
in Intelligent Tutoring Systems

Michael J. Baker
B.A.(Hons), DipMus

Submitted for the degree of
Doctor of Philosophy in Cognitive Science

Research conducted in the Institute of Educational Technology,
Centre for Information Technology in Education,
The Open University, Milton Keynes, Great Britain

July 1990

© Michael Baker

Author's number : M 7023835

Date of submission : 29th September 1989

Date of award : 13th July 1990

BEST COPY

AVAILABLE

Variable print quality

Abstract

This thesis describes a general approach to tutorial interaction in Intelligent Tutoring Systems, called "Negotiated Tutoring". Some aspects of the approach have been implemented as a computer program in the 'KANT' (Kritical Argument Negotiated Tutoring) system. Negotiated Tutoring synthesises some recent trends in Intelligent Tutoring Systems research, including interaction symmetry, use of explicit negotiation in dialogue, multiple interaction styles, and an emphasis on cognitive and metacognitive skill acquisition in domains characterised by justified belief. This combination of features has not been previously incorporated into models for intelligent tutoring dialogues.

Our approach depends on modelling the high-level decision-making processes and memory representations used by a participant in dialogue. Dialogue generation is controlled by reasoning mechanisms which operate on a 'dialogue state', consisting of conversants' beliefs, a set of possible dialogue moves, and a restricted representation of the recent utterances generated by both conversants. The representation for conversants' beliefs is based on Anderson's (1983) model for semantic memory, and includes a model for dialogue focus based on spreading activation. Decisions in dialogue are based on preconditions with respect to the dialogue state, higher level educational preferences which choose between relevant alternative dialogue moves, and negotiation mechanisms designed to ensure cooperativity.

The domain model for KANT was based on a cognitive model for perception of musical structures in tonal melodies, which extends the theory of Lerdahl and Jackendoff (1983). Our model ('GRAF' - GRouping Analysis with Frames) addresses a number of problems with Lerdahl and Jackendoff's theory, notably in describing how a number of unconscious processes in music cognition interact, including elements of top-down and bottom-up processing. GRAF includes a parser for musical chord functions, a mechanism for performing musical reductions, low-level feature detectors and a frame-system (Minsky 1977) for musical phrase structures.

Dedicated
with love
to my parents,
David and Marlene Baker

Acknowledgements

I would like to express my warmest thanks to my supervisor, Dr. Mark Elsom-Cook, for his willingness to bend his mind towards my problems, for his help, criticism and inspiration over the past three years, and for reading my first draft very quickly.

I thank Tim O'Shea for providing motivation and opportunities to gain experience as an academic and researcher in the middle period of this research. The research group at the Centre for Information Technology in Education is a very motivated and 'close-knit' one. Nearly all its members have therefore helped me over the past three years - many thanks, it has been a pleasure to work with you all. A special mention for the following members of the group, present and departed:

Alistair Edwards, Pat Fung, my contemporary and friend who 'took me in' in the early days, Simon Holland, Rick Evertsz, Eileen Scanlon for listening to me and 'bailing me out', Fiona Spensley, who read innumerable earlier papers and helped with early programming problems, Claire O'Malley, who put some of my psychology right, Richard Joiner, Rod Moyse, Laurence Alpay.

A special thankyou to Ann Blandford, for taking an interest in my research, and for exemplary proof-reading at the last minute.

A special thanks to the administrative and secretarial staff in I.E.T. - Olwyn Wilson, Di Mason, Kathy Edwards, Pat Cross - who were always ready to help out. Olwyn Wilson helped me to get a much needed overseas grant to speak at AAAI in the States in 1988, for which I can't thank her enough. I couldn't have finished this thesis without technical support from John Close and Dave Perry.

Thanks to all my friends in the OU Human Cognition Research Laboratory who provided help and support of many kinds, to John Domingue, Tim Rajan, Tony Hasemer, Mike Brayshaw, and to Arthur Stutt for discussing research with me.

A number of researchers in Universities in England and Europe have helped to shape this research. My thanks to Anders Friberg, Eric Clarke, Peter Howell, Ian Cross, Philip Johnson-Laird, Steven McAdams, Fred Lerdahl, Alison Petrie-Brown, John Self and Ben DuBoulay.

All my love to Mido for persuading me to carry on, for giving me the ability to finish, and a reason to finish. Thanks to Genevieve for being who she is.

Thanks to Mark, Jo and Bertie for their friendship and for helping me to remain a human being.

Thanks to Susan for many things.

Finishing this research on time has necessitated prolonged absence from family and friends. I dedicate this work to my parents and send them all my love and thanks for tolerating my neglect, and for always supporting me in whatever I've done, in whatever decisions I've made: in many ways I did it for them.

This research was supported by a postgraduate award from the Science and Engineering Research Council of Great Britain.

Contents

Chapter 1: A thesis in Intelligent Tutoring Systems	1
0.0 A thesis in Intelligent Tutoring System	1
1.0 Aims of the thesis	1
1.1 An approach to ITS research	1
1.1.1 Modelling the interaction	
1.1.2 The methodology of cognitive simulation	
1.1.3 ITS as interdisciplinary research	
1.1.4 Domain models and dialogue models	
1.2 A cognitively based domain model for musical structures	5
1.3 Negotiated Tutoring as an approach in ITS research	6
1.3.1 Interaction symmetry	
1.3.2 Negotiation and cooperation in dialogue	
1.3.3 Multiple symmetrical interaction styles	
1.3.4 Cognitive and metacognitive skills	
1.4 Origins of the thesis	9
1.5 Structure of the thesis	10
 Chapter 2: Cognitive Models for Musical Structure	 11
2.0 Introduction	11
2.1 Factors relating to the occurrence of musical structures	11
2.2 Existing work relevant to analysis of musical structures	13
2.2.1 Musical patterns	
2.2.2 Harmonisation and musical grammars	
2.2.3 Lerdahl and Jackendoff's "Generative Theory of Tonal Music"	
2.2.3.1 Grouping structure	
2.2.3.2 Time-span reductions	
2.2.3.3 Conclusions concerning Lerdahl & Jackendoff's theory	
2.2.4 Approaches to refining the theory of Lerdahl and Jackendoff	
2.2.5 Summary and prospects for a cognitive theory of musical structure	
2.3 Towards a Cognitive Model for Musical Phrase Structures	21
2.3.1 Introduction	
2.3.2 Overview of AGA and GRAF	
2.4 AGA: a syntactic harmonic processing approach	22
2.4.1 The aim of the model	
2.4.2 The Automated Grouping Analysis System	
2.4.2.1 From the musical surface to the musical lexicon	
2.4.2.2 The parsing process	
2.4.2.3 The reductional process	

2.4.2.4 Deep reductional parallelism	
2.4.2.5 Grouping at the sub-phrase level	
2.4.3 Preliminary assessment: psychology and syntactic processing of music	
2.5 A Frame-based Approach to Grouping Analysis	36
2.5.1 Introduction	
2.5.2 Recognition with schemata and frames	
2.5.3 Design for a frame-based parser: the GRAF system	
2.5.4 An example grouping recognition with the GRAF model	
2.5.5 Summary of GRAF	
2.5.6 Limitations of knowledge-based approaches	
2.5.7 Critical assessment of syntactic and recognition-based approaches	
2.6 Summary and conclusions	49
2.6.1 Research contributions	
2.6.2 GRAF as a domain model for Intelligent Tutoring	
 Chapter 3: Computational Approaches to Modelling High-level Human-computer Dialogue	 52
3.0 Introduction	52
3.1 Approaches to Interaction in Intelligent Tutoring Systems	52
3.1.1 Socratic dialogues: SCHOLAR, WHY and TRILL	
3.1.2 Semantic grammars: SOPHIE	
3.1.3 Coaching, issues and examples: WEST	
3.1.4 Discourse Management Networks: MENO-TUTOR	
3.1.5 Case method dialogues: GUIDON	
3.1.6 Dialogue and Guided-Discovery Tutoring: IMPART	
3.1.7 Summary of approaches to interaction in ITS	
3.2 Explanation and Dialogue	70
3.2.1 Explanation in rule-based systems	
3.2.2 Explanation and interactive dialogues	
3.3 Argument and Dialogue	76
3.3.1 Reasoning, explanation, argument and dialogue	
3.3.2 Approaches to generation and understanding of argument	
3.4 Computational Models of Dialogue	82
3.4.1 Introduction	
3.4.2 Dialogue grammars and schemata	
3.4.3 Topic structures in dialogue	
3.4.3.1 Focus and task-oriented dialogues	
3.4.3.2 Context spaces and conversational moves	
3.4.4 Topic and Intention in Dialogue	
3.4.4.1 Topic, Intention and Attention in Dialogue	
3.4.4.2 Dialogue games	

3.4.4.3 Summary: Topic and Intention in Dialogue	
3.4.5 Dialogue, Goals and Planning	
3.4.5.1 Planning Cooperative Dialogues	
3.4.5.2 Inferring Goals and Plans in Dialogue	
3.4.5.3 Planning Utterances to Achieve Multiple Goals	
3.4.6 Computational Models of Dialogue : Summary	
3.5 Conclusion	113
 Chapter 4: Negotiated Tutoring - an approach to intelligent tutorial interaction	115
4.0 Introduction	115
4.1 Models of high-level dialogue	115
4.2 Primacy of the interaction model	116
4.3 Research methodology	117
4.4 Design goals: Negotiated Tutoring	118
4.5 Negotiation, symmetry and cognitive skills in dialogue	123
4.6 Negotiated Tutoring features in the example dialogue	131
4.7 Knowledge and belief in dialogue	135
4.8 Summary of design goals in Negotiated Tutoring	136
 Chapter 5: Specification and implementation of the Intelligent Tutoring System KANT	139
5.0. Introduction	139
5.0.1 An Overview of the Architecture of KANT	
5.1 Representing the Dialogue State	142
5.1.1 The notion of a 'dialogue state'	
5.1.2 The Domain knowledge and its Derivation	
5.1.3 System and student belief representations	
5.1.3.1 The semantic network representation	
5.1.3.2 Anderson's theory of factual memory	
5.1.3.3 Use of Anderson's theory in KANT	
5.2 The dialogue history	149
5.3 Dialogue goal operators	151
5.3.1 The general structure of dialogue moves	
5.3.2 The hierarchy of dialogue moves	
5.3.2.1 Making claims	
5.3.2.2 Making challenges	
5.3.3 Dialogue moves and teaching styles	
5.3.4 Origin of the dialogue moves	
5.4 An overview of dialogue generation mechanisms	164
5.4.1 The Fundamental Algorithms	
5.4.1.1 The ARGUMENT CONTROLLER	

5.4.1.2 The CONVERSER	
5.4.1.3 Discriminating between alternative dialogue subgoals	
5.4.1.4 Conversing a single dialogue goal	
5.4.1.5 Variants of the dialogue generation programs with the student as negotiator	
5.5 Some components of the dialogue model in more detail	176
5.5.1 Dialogue State Preconditions	
5.5.1.1 Examples of dialogue state preconditions	
5.5.1.2 Preconditions for determining focus, concepts and instances	
5.5.2 Dialogue Negotiation	
5.5.2.1 The system as negotiator	
5.5.2.2 The student as negotiator	
5.5.2.3 Negotiation, acceptance and rejection	
5.5.3 Dialogue Actions	
5.5.3.1 System INFORMing	
5.5.3.2 Student informing	
5.5.3.3 The fundamental algorithm for the INFORMER	
5.5.3.4 Example action	
5.5.3.5 Summary of dialogue actions	
5.5.4 Action Effects and Negotiation Effects	
5.6 Example Dialogues and their Generation	207
5.7 Extensions to the present implementation towards the present specification	215
5.8 Summary of Chapter 5	217
Chapter 6: Conclusions, critique and further work	218
6.0 Introduction	218
6.1 Critique	218
6.1.1 A cognitive model for musical structures: critique	
6.1.2 KANT and the Negotiated Tutoring approach: critique	
6.2 Research contributions	223
6.2.1 ITS research: contributions	
6.2.2 Education theory: contributions	
6.2.3 The cognitive science of music: contributions	
6.2.4 Computational musicology: contributions	
6.3 Further work	228
6.3.1 The cognitive model for musical structures: further work	
6.3.2 Extending KANT towards Negotiated Tutoring: further work	
6.3.3 Integrating GRAF and KANT: further work	
6.3.4 The Negotiated Tutoring approach: further work	
6.4 Conclusion	235

References

236

Appendices

Appendix A: Architecture of an ITS for Musical Structure and Interpretation

Appendix B: Semantic Network for Musical Structures

Appendix C: Simplified Set of Dialogue Moves used by KANT

Table of Figures

Chapter 2	11
Figure 2.0: Example grouping analysis of theme from Mozart's G minor symphony	
Figure 2.1: The 'normal form' for the interrupted form	
Figure 2.2: A section of the time-span reduction from the chorale "O Haupt voll Blut und Wunden", (St. Matthew Passion), J.S. Bach	
Figure 2.3: Harmonisation of the first two bars of Mozart piano sonata in C, K545	
Figure 2.4: A simplified algorithm for performing musical reductions	
Figure 2.5: An example reduction of the first phrase of Mozart K331	
Figure 2.6: Harmonic formats for phrase structures of simple nursery rhyme melodies	
Figure 2.7: The relationship between grammar rules and harmonic formats	
Figure 2.8: Diagram of frame system for harmonic phrase formats	
Figure 2.9: A simplified algorithm for the top-level control structure of the frame-based musical grouping recogniser 'GRAF'	
Figure 2.10: A discrimination net for the four possible phrase formats in the context of English Nursery rhyme melodies	
Figure 2.11: Hand-simulated example of frame-based grouping recognition	
Chapter 3	52
Figure 3.1: The BLAH system (Weiner 1980)	
Figure 3.2: An example argument according to Toulmin et al's (1979) argument analysis	
Figure 3.3: Hierarchy of linguistic actions, described by Appelt (1983)	
Chapter 4	115
Figure 4.1: A simplified geological map, showing positions where rock specimens were collected	
Figure 4.2: Part of a semantic network for knowledge and beliefs of dialogue participant A	
Chapter 5	139
Figure 5.0: The architecture of KANT	
Figure 5.1: A portion of the semantic network linking sources of justifying knowledge	
Figure 5.2: Example dialogue goal operator for making a CLAIM	
Figure 5.3: A simplified dialogue goal hierarchy	
Figure 5.4: An example high-level dialogue structure of negotiated turns and topic shifts	
Figure 5.5: Example subconcept network, beneath the concept 'harmony'	

Chapter 1:

A thesis in Intelligent Tutoring Systems

0.0 A thesis in Intelligent Tutoring Systems

Our thesis is that

Negotiated Tutoring is a model for the generation of teaching interactions in Intelligent Tutoring Systems, which indicates how some aspects of recent trends in intelligent teaching dialogue and human-computer dialogue research may be implemented in computer programs.

1.0 Aims of the thesis

The aim of this thesis is to develop a general theoretical model for intelligent educational interaction called *Negotiated Tutoring*. Negotiated Tutoring aims to synthesise a number of recent trends in Intelligent Tutoring Systems ('ITS') research within a coherent theoretical model. The trends include increased symmetry in the range of interactions available to the tutoring system and the student, the synthesis of multiple interaction styles in terms of lower level dialogue units, the use of explicit negotiation of cooperative interaction goals between system and student and an emphasis on cognitive and metacognitive skill acquisition. The model is instantiated within an ITS for musical structures, called 'KANT'. Research focussed on two design and implementation issues:

1 a cognitive domain model ('GRAF') for perception of musical structures, intended as a domain model for KANT;

2 a model for tutorial interaction called *Negotiated Tutoring*, instantiated in the ITS 'KANT'. The system generates tutorial dialogues based on critical arguments with an educational purpose, which is specifically adapted to the characteristics of uncertain justified belief in the knowledge domain of musical structures.

Our principal research contribution concerns the development of the Negotiated Tutoring approach to interaction in Intelligent Tutoring Systems, which is implemented within KANT. The system generates tutorial dialogues within the domain of musical structures.

1.1 An approach to ITS research

ITS research is now entering a phase of development where the 'traditional trinity' (Self 1989) of 'domain, student model and teaching strategies' is beginning to be questioned, along with its attendant emphasis on domain and student modelling, formalisation of teaching rules and knowledge communication. We therefore need to clearly state the general approach to ITS research which we have adopted in this thesis.

1.1.1 Modelling the interaction

The general approach adopted in this thesis is to design a model for educational interactions embodied in the design and partial implementation of a computer program which will support a stand-alone one-to-one interaction between a human student and a computer. This approach is grounded in an ITS paradigm, well established in the field¹, which aims to provide individualised and adaptive instruction, and is currently being complemented by research in computer-supported collaborative work. A wider perspective on this ITS approach comes from the field of human-computer interaction, where teaching interactions can be seen as a special case of high-level dialogue between rational agents (Kiss 1986; Cohen & Levesque 1987). Earlier ITS research had focussed on the development of models of the teaching domain (Brown, Burton & DeKleer 1982), and the modelling of student knowledge in terms of domain representations (Burton 1982). In common with a current tendency in ITS research (Elsom-Cook 1984, 1989a, 1989b) we focus on modelling *the interaction*, as our primary source of evidence concerning the teaching process. A fundamental premise of our approach is that the interaction problem can be considered in terms of the development of formal models of *high-level dialogue*, considered in separation from the specific low-level media (text, graphical displays, sounds) in which higher level structures and processing mechanisms in dialogue generation are expressed.

1.1.2 The methodology of cognitive simulation

Our approach to the development of such interaction models (and of models of the teaching domain) is based on the methodology of *simulation* in cognitive science. An initial definition of computer simulation might be that it aims to instantiate and test a theory of human cognition, whereas an artificial intelligence ('AI') program attempts to simulate behaviours which would be considered intelligent on the part of human beings, without necessarily claiming that the programs do this in a humanlike manner. In practice, the AI/cognitive simulation distinction is often (fruitfully) blurred:

"... an AI system may reflect psychological theories when these happen to suggest a feasible way to do the job. AI may also yield psychological insights in the course of developing and refining a system"

(Cohen 1983, p. 194).

An advantage of using computer simulation as a means of developing theories, is the adoption of a methodological criterion of 'implementability', to enable theories to be precisely and completely specified in the form of computer programs and their formal specification. At present, there are few existing theoretical models of the processes underlying dialogue comprehension and generation² which can be applied to the development of models of the

¹ The approach is summed up in the relatively early collection of articles in Sleeman & Brown (1982).

² Exceptions being the work of Power (1979), Grosz & Sidner (1986), Appelt (1982) and Cohen & Levesque (1987).

teaching interaction, and formal models of the subject domain which we considered are presently in their infancy. Our approach has therefore been to attempt to integrate and reformalise existing work. In terms of modelling the teaching interaction, this included work in education theory on teaching strategies, cognitive psychology of memory, artificial intelligence models of rational agency, and studies of dialogue structure in computational linguistics. We do not believe that theory can be inferred 'bottom-up' from data (in our case on teaching dialogues) without some 'prototheory' which we are trying to test³. We therefore adopted the view that a 'synthetic' approach to developing such a 'prototheory' was appropriate, given the present paucity of formal theories of dialogue and the teaching interaction. Similarly, in the field of formalising our teaching domain, we took existing informal theories (such as that of Lerdahl & Jackendoff 1983), and attempted to reformulate and extend them, with reference to existing theories in cognitive science, and results in the experimental psychology of music. In summary, this research can be viewed as theoretical cognitive science applied to Intelligent Tutoring Systems.

1.1.3 ITS as interdisciplinary research

The field of ITS is defined by its highly *interdisciplinary* nature. We therefore believe that the primary function of ITS research should be that of *synthesis* of research in contributing disciplines. However, contributions from those disciplines (such as in education, and music theory) are very often not sufficiently formalised for suitable use in an ITS, simply because these disciplines do not usually pose research questions in exactly the manner of ITS research, and do not seek similarly formalisable answers to them. ITS does, therefore, have something to offer to these other disciplines in terms of the demands of formalisation and extension of theories⁴. The research described in this thesis has implications for

- *educational research*, in terms of the formalisation of teaching strategies, and of educational principles for choosing between alternative strategies;
- *computational linguistics, and artificial intelligence* in terms of formal models for generation of dialogue;
- *the cognitive science of music*, in terms of a model for perception of musical structures;
and
- *ITS research*, in terms of a high-level model for generation of critical argument dialogues, and the synthesis of recent research aims in the Negotiated Tutoring approach.

There is a further aspect to the interdisciplinary nature of ITS research: the field is now sufficiently developed that research can focus on specific problems, such as interaction models, teaching strategy formalisation, aspects of student modelling and machine learning, multiple viewpoints in knowledge representation, and so on. We therefore believe that '*whole systems*'

³ See Hempel, C.G. (1966), *Philosophy of Natural Science*, Prentice-Hall, N.J.

⁴ Self, quoting Clancey (1986) states that

"... ITS is not a 'mere matter of putting well-known AI methods into practice' ... ITS is (or should be) 'broadening the meaning of AI research'..."
(Self 1989, p.1).

ITS research is no longer appropriate (Self 1988). However, due to the close interrelationships between individual problems, we believe that any ('ITS') solution to individual problems must be derived with some more general conception of how they could *integrate* within a whole system architecture, in order to address a real educational problem. For this reason we have focussed research on two fundamental and closely related problems: domain models and dialogue models.

1.1.4 Domain models and dialogue models

We focussed specifically on the two issues of modelling the teaching domain and modelling the structure of the interaction because of their clear interrelationship : the representation of knowledge influences how we can talk about it, and interaction makes demands on knowledge representation. Existing applications of artificial intelligence in education have generally been restricted to domains which are already highly formalised, such as branches of mathematics, physics and computer science (Sleeman & Brown 1982). This initial approach soon faced the problem that existing explicit formalisations for domains and their respective reasoning methods did not necessarily correspond to the way in which *humans* represented and reasoned in these domains, and thus did not form appropriate representations for modelling the knowledge which was intended to be communicated to a student. Examples include the initial research on the ITS 'SOPHIE' (Brown, Burton & DeKleer 1982), which began with mathematical models for reasoning in electronics and led to research on 'mental models', and the ITS GUIDON (Clancey 1982, 1987), which demonstrated the unsuitability for tutorial communication of expert system knowledge bases as a 'deep' cognitively based domain representation⁵. A further assumption of the approach of taking pre-existing formalisations and the attendant artificial intelligence knowledge representations of the day, was that ITSs were endowed with knowledge which was *complete* within a given domain, and which was assumed *a priori* to be 'correct' or certain. An initial motivation of this thesis was therefore to explore the consequences for intelligent tutoring of attempting to design an intelligent tutoring system in a domain for which there was *no preexisting complete formalisation*, for which *cognitive models were little developed*, and where 'experts' in the domain recognised that knowledge was largely *intuitive, uncertain*, and intrinsically admitting of *multiple possible solutions*.

The knowledge domain which we chose for exploring these issues was that of the analysis of musical structures (as described in chapter 2). In choosing this subject area, it was recognised that its specific characteristics are to a greater or lesser extent present in many other domains: music was chosen because the field simply leaves us with no choice but to confront these issues. The lack of existing formal cognitive models for most areas of music forces us to reconsider afresh the problem of representing a domain for tutoring interaction. As we shall describe (see chapters 4 & 5), the Negotiated Tutoring approach runs counter to the general idea of ITSs as primarily (or exclusively) concerned with *knowledge communication* (Wenger 1987). In the

⁵ Witness the emphasis of Clancey's recent work (1988) on graphical interfaces and 'qualitative models'.

research reported in this thesis, there were two main results derived from considering the relationship between domain and dialogue models:

1 In choosing a cognitive modelling technique for perception of musical structures, from the available possibilities in this unformalised domain, we excluded approaches which were not suitable as the basis for tutorial interactions. For example, connectionist and 'blackboard' models were rejected since they may produce acceptable solutions (to the problem of identifying positions of structural boundaries in music), but are essentially 'black box' (Du Boulay, O'Shea & Monk 1981) models, which give no basis for *explaining* the rationale for these solutions to a student. The approach chosen (described in chapter 2 of this thesis) does justice to the cognitive characteristics of the domain of analysis of musical structures, in terms of the inherent *uncertainty* of putative structures, and the *ambiguity* amongst multiple possible solutions. We therefore adopted a *plausible reasoning technique*, using a frame-based parser.

2 Given that knowledge in this domain is uncertain, directive interaction styles which aimed to inculcate these uncertain beliefs in the student were necessarily precluded. We therefore adopted a set of fundamental dialogue units (called "dialogue moves", see chapter 5) which aimed to acquaint the student with the range of possible beliefs and their justifications in the domain, and to teach the necessary domain concepts required to understand them. The dialogues generated are termed *critical arguments with an educational purpose*. Given the range of dialogue moves available to the system, relating to a set of justified beliefs, it seems natural that a more cooperative and symmetrical style of tutoring is appropriate, where the system does not simply aim to communicate its beliefs without allowing the student a similar possibility to interact. Negotiated Tutoring thus arises naturally from the cognitive characteristics of this domain. We should emphasise that Negotiated Tutoring is a general *approach* to intelligent tutoring, and could equally well be implemented with a different set of dialogue moves, which possibly aimed to communicate knowledge which was assumed to be certain and complete: it is simply that the egalitarian and cooperative nature of the Negotiated Tutoring approach naturally coheres with the dialogue moves of critical arguments and the goals of tutoring with justified belief. It is clear that the possession of justified belief by a student or teacher is a general cognitive characteristic which may recur throughout a number of the areas of knowledge which any particular culture isolates as a 'subject' or 'domain'.

1.2 A cognitively based domain model for musical structures

One approach to developing a domain model for an ITS which aimed to teach a student about musical structures would be to simply prestore declarative knowledge of musical structures for a predefined set of example melodies. We did not adopt this approach since the set would be necessarily finite, and such a representation would provide little pedagogical leverage in a tutorial interaction for explaining *why* such structures existed in given examples, thus making it unlikely that a student would be able to generalise this knowledge to new examples. We

wanted to explore the possibility of developing a model for automated analysis, which could *derive* musical structures for input melodies, and so be of wider educational application and provide a basis for explaining such structures in terms of the processes which led to their derivation. Since the reasoning processes involved would be intended to be communicated to a student, this implied that a *cognitive model* for the domain was required. The development of such a model enabled us to consider the needs of the interaction in terms of choosing a suitable domain modelling technique. AI approaches which depended on reasoning techniques which were largely statistical or numerical (approaches to 'reasoning with uncertainty' (Cohen 1985) and connectionist approaches (Bharuca 1987; Lischka 1987)) were therefore rejected since they would not form a reasonable basis for a tutorial interaction. The method finally chosen attempted to do justice to the characteristics of intrinsic *uncertainty* and *ambiguity* in the musical structures in many pieces, which is well established by musicians and psychologists of music⁶, and to the findings of experimental psychology of music that high-level schemata are involved in musical grouping perception (Stoffer 1985; McAdams 1987). We therefore chose a method based on a frame-based recognition, integrated with a harmonic parser, which uses *plausible reasoning techniques* and which would form the basis for explaining its set of possible solutions in terms of the musical factors used in their derivation (including factors such as harmony, and musical 'surface' features such as Gestalt factors). The approach integrates a musical parser of chord functions - since phrase boundaries relate to harmonic progressions - as a *discrimination test* between possible matching grouping schemata, suggested by features on the musical surface. It is restricted to analysis of tonal melodies of a particular genre, and its domain restrictions have implications for the kind of tutoring which the dialogue model can perform. The solutions derived from the model are considered as a set of *justified beliefs*, represented as *concept instances* in a semantic memory representation of the musical concepts required to explain the nature of these beliefs and the knowledge sources of their justifications. Tutorial communication of these beliefs does not simply aim to impart them as generated 'templates' but rather to integrate their educational use within a coherent and structured tutorial dialogue. Their status as justified belief has implications for the dialogue model in terms of the kinds of 'dialogue moves' which can be used, and suggests the cooperative tutorial style of Negotiated Tutoring.

Apart from the interest of the domain of musical structures for ITS research, we suggest that the work described in this thesis may have an important contribution to make in the field of music education - particularly in teaching the high-level skill of musical interpretation. These issues are strictly beyond the scope of this thesis, but Appendix A contains a speculative proposal concerning the application of this research to teaching musical interpretation.

1.3 Negotiated Tutoring as an approach in ITS research

Negotiated Tutoring is a general model of Intelligent Tutoring, which has its roots in a number of current tendencies in ITS research. Negotiated Tutoring synthesises four main features:

⁶ See Lerdahl & Jackendoff (1983); Sloboda (1985).

- 1 interaction symmetry
- 2 use of negotiation to secure cooperation in dialogue
- 3 multiple interaction styles
- 4 cognitive and metacognitive skill acquisition

We shall briefly motivate each of these features in turn.

1.3.1 Interaction symmetry

A current tendency in ITS research concerns a shift away from the view of the tutoring system as coercively constraining the student to be simply the passive respondent of its pedagogical decisions. The converse of this situation arose in coaching systems (such as 'WEST', Burton & Brown 1982) which guided a student's exploration in some computer-based environment, but where (suitably constrained) interactions were still largely confined to the system, and the student had little opportunity to direct the interaction towards their own learning goals. In either case, the structure of the educational interaction is strongly *asymmetrical*, in comparison with the emphasis of 'progressive' teaching styles⁷. Such 'progressive' styles aim to treat learners as active participants in their own learning, and thus give greater freedom and opportunity to interact in pursuit of their own learning goals, which are not defined as the simple inverse of the teacher's strategies. Negotiated Tutoring therefore aims to emphasise *increased symmetry* in the teaching interaction, as far as possible given the current state of human-computer interaction research, in terms of the *range* of interaction styles available to both dialogue participants, and the *freedom* and opportunity to *negotiate cooperative pursuit* of interaction goals. In the KANT system (chapter 5 of this thesis) this symmetry is basically provided by considering the dialogue as a succession of *negotiated turns*, which are available to either participant to negotiate a common set of interaction styles, defined as a succession of lower-level 'dialogue moves'. In KANT the set of dialogue moves represented correspond to those required to generate the *critical arguments* which are appropriate to the cognitive characteristics of the domain considered - i.e. uncertain justified belief.

1.3.2 Negotiation and cooperation in dialogue

An emphasis on *negotiation*, *cooperation* and *freedom* is a further feature of Negotiated Tutoring, which emerges as a consequence of *symmetry*: if both participants in the dialogue have equal (symmetrical) freedom to pursue interaction goals, and neither can or should *coerce* the other, then cooperation is required for mutual satisfaction of these goals, and negotiation is one way to achieve this. These features have strong affinities with some of the goals of the 'social cognition' group, concerned with human-computer interaction research in general, and Seely-Brown's (1989) 'new epistemology of learning'. Their work emphasises a number of other

⁷ Bennett (1976) is the classic discussion of the distinction between 'progressive' and 'traditional' teaching approaches, and their relative effectiveness, in which progressive approaches emphasise the following features: 'Active pupil role', 'Pupils participate in curriculum planning', 'Teacher as guide to educational experiences' (p. 38). Traditional approaches emphasise the converse of these features.

features of cognition and learning, including its necessary placement 'in situ' in 'everyday learning'.

1.3.3 Multiple symmetrical interaction styles

A current in ITS research concerns a move towards providing a range of *teaching styles* (Goodyear 1989, in press) and an attempt to develop educational principles for deciding which is the most appropriate at any given point in the interaction (Elsom-Cook & Spensley 1988). Multiple teaching strategies are required if an ITS aims to be fully adaptive to the range of multiple learning styles preferred by students. The Negotiated Tutoring approach emphasises that just as single teaching styles should not be viewed as the inverse of single learning styles, the same applies to *multiple* teaching styles. Any solution to the problem of choosing between alternative teaching strategies at a given point in the interaction, depends crucially on the *theoretical definition* of teaching strategies. In chapter 4 we argue that teaching styles should not necessarily be viewed as fixed pairings of limited interaction units with a single domain traversal mechanism, but that at some point teaching styles must be represented as flexible successions of *dialogue moves*, flexibly paired with multiple possible domain traversal methods. A further feature of Negotiated Tutoring is therefore the emphasis that at some point research in teaching styles and dialogue must be closely linked. In KANT multiple interaction styles (teaching and learning) are generated via the provision of a set of *dialogue moves* which can be flexibly combined within an interaction using a mechanism for *explicit negotiation*.

1.3.4 Cognitive and metacognitive skills

An important feature of Negotiated Tutoring concerns an emphasis on the acquisition of *cognitive and metacognitive skills* as a goal for ITS, in distinction with the prevalent view of ITS as essentially or primarily concerned with "knowledge communication" (Wenger 1987). As Self points out,

"Such an 'ITS philosophy runs counter to almost everything of significance in twentieth century educational philosophy"
(Self 1989)

As part of Negotiated Tutoring we incorporate the view that ITS should be concerned with the cognitive skill of *reasoning in dialogue*, which is not identical with reasoning as a mental cognitive process (see chapters 3 and 4 of this thesis), and which we term critical argument. The purpose of Negotiated Tutoring is therefore to facilitate the acquisition of this skill which is essentially grounded in *dialogue itself*, by the very process of engaging the student in such a dialogue which displays the 'target' cognitive skill. We claim that KANT is a simplified first prototype which indicates a research program for generating such tutorial dialogues. Given the cognitive skill of 'reasoning-in-dialogue', Negotiated Tutoring also aims to facilitate the acquisition, and promote the use of, the *metacognitive skill* of *belief revision*. It does this via

the specific characteristics of the set of dialogue moves for critical argument provided, where the aim is to allow the student to reflect on their own set of beliefs in the domain by the explicit discussion of their beliefs and their justifications with those of a second dialogue participant in a structured educational interaction. In a sense, we can therefore view the dialogue as a formalisation and extension of many of the goals of 'Socratic tutoring'⁸ (Collins & Stevens 1983). There are also close affinities with recent work on formalisation of theories of rational agents using logics of belief and intention (Kiss 1986; Cohen & Levesque 1987).

We should emphasise that *Negotiated Tutoring is not identical with critical argument dialogues*: it would be possible to substitute a set of dialogue goals which aimed to communicate declarative 'certain' knowledge into KANT, and retain the essential symmetrical and negotiative features of Negotiated Tutoring. However, the requirements of the Negotiated Tutoring approach are particularly apparent in the case where an ITS possesses justified belief, and should therefore not aim to produce a highly directive and coercive teaching style in an attempt to communicate these beliefs. The idea of justified belief lends itself to an egalitarian conception of educational interactions where participants are considered as equals. We believe that Negotiated Tutoring should be considered in terms of a set of current approaches in ITS and the philosophy of education which *naturally cohere*. In chapter 4 of this thesis we expand on the nature of the relationship between the principal elements of Negotiated Tutoring.

1.4 Origins of the thesis

The cognitive model for musical structures described in chapter 2 was originally inspired as an attempt to reformulate the theory of Lerdahl and Jackendoff (1983), which is a dominant theory in the field of music and the cognitive sciences, and benefitted from discussions with Lerdahl in 1987-88. Further major influences were the work of Steedman (1984) on grammars for twelve-bar blues chord progressions, and discussions with Philip Johnson-Laird at Cambridge University in 1988, Eric Clarke (see Clarke 1985) at the Second International Conference on Science and Music, City University, 1986-89, Stephen McAdams (see McAdams 1987) at the First International Symposium on Music and the Cognitive Sciences at IRCAM (Paris) 1987-89, and the work of the psychologist of music John Sloboda (see Sloboda 1985).

The view of dialogue and interaction as the primary issues of importance in ITS was strongly influenced by the work of Elsom-Cook (1984; 1987; 1989a; 1989c). Negotiated Tutoring as an approach relates to work on cooperative explanation by O'Malley (1987b), the work of Stutt (1987) on argument in expert systems, and by an attempt to reformulate work on critiquing (Miller 1984) and Socratic Tutoring (Collins & Stevens 1983; Cerri, Elsom-Cook & Leoncini 1988) and integrate it within dialogue frameworks. In the field of dialogue *per se* the influences were numerous and evident, particularly the work of Levin & Moore (1977), Power (1979), Elsom-Cook (1984), and theories of plans and agents (Appelt 1982; Cohen & Perrault 1979; Cohen & Levesque 1987). The work of Self (1988, 1989) helped to crystallise these ideas into

⁸ Elsom-Cook has formalised many of these dialogue goals, in a recent unpublished paper, to appear in Elsom-Cook (ed.) (1989).

the Negotiated Tutoring approach.

1.5 Structure of the thesis

The central chapters of this thesis are concerned with cognitive modelling for the domain of musical grouping structures, and an approach to the generation of tutorial dialogues for that domain. The plan of chapters can be summarised as follows:

Chapter 2: A cognitive model for musical grouping structures describes a new model for perception of musical grouping structures, based on a frame-based parser. It is intended as a domain model for the dialogue model described in subsequent chapters.

Chapters 3 to 5 are concerned with models of the tutorial interaction. The Negotiated Tutoring approach is presented, with an implementation of some aspects of the approach in an ITS.

Chapter 3: Computational approaches to modelling high-level human computer dialogue reviews computational approaches to human-computer dialogues, including research in ITS, explanation and argument, computational linguistics and AI models of rational agents.

Chapter 4: Negotiated Tutoring presents the theoretical basis of an approach to intelligent tutoring. We describe how the approach synthesises a number of current tendencies in ITS research within a coherent framework which is computationally tractable.

Chapter 5: Specification and implementation of KANT gives a complete description of a design and prototype implementation of an ITS called KANT, which incorporates some aspects of the Negotiated Tutoring approach.

Chapter 6: Conclusions, critique and further work summarises conclusions and research contributions from previous chapters, critiques KANT in relation to existing dialogue and ITS research, and summarises further work.

Chapter 2: Cognitive Models for Musical Structure

"... to really understand how memory and process merge in "listening" we will simply have to use much more "procedural" descriptions - that is, the kinds that can describe how processes proceed."

(Marvin Minsky, *Music, Mind and Meaning*, 1982, p.6)

"We have not developed a theory of the algorithms used in the processing of music ..."

(Ray Jackendoff, *Consciousness and the Computational Mind*, 1987, §11.9)

2.0 Introduction

In this chapter we review cognitive science approaches to the analysis of musical structure, from the point of view of determining a *domain model* for the tutorial dialogue model described in the remaining chapters of the thesis. In the last section we propose our own model for perception of musical grouping structures, based on a reformulation of the theory of Lerdahl and Jackendoff (1983). The model was partially implemented as a computer program, and unimplemented features were specified. In conclusion we discuss how properties of our cognitive model for the domain of musical structure relate to the dialogue model used to discuss it.

The general question which we seek to answer is

how can a computer program be specified which models how listeners perceive musical phrase boundaries in unharmonised melodies?

As will emerge from our review, no existing work has posed this question, having mostly adopted the simplifications that either a full harmonisation or an indication of harmonic boundaries of a melody is already given as input to a program. Since we were interested in exploring the relationship between unconscious inference of harmony and perception of phrase boundaries, we did not make this simplification. We did, however, make the simplification that *metre* was an input to the program. Our question was thus restricted to considering the problem of specifying a computer program which would take a list of note representations, metrical boundaries, the key of the melody and a set of knowledge bases for a harmonic grammars and lexicon as input, and produce a set of possible phrase boundaries as output. We were therefore not concerned with computer analysis from the level of acoustical input, and note representations were input into the program by hand.

2.1 Factors relating to the occurrence of musical structures

Given our goal of specifying an algorithm for analysis of musical phrase boundaries, we need to describe which features relate to their occurrence. The principal unit of structure in Western tonal music is the *phrase*. It is not easy to find precise definitions of what a 'phrase' is in the musical literature. Schoenberg thus simply defines a phrase as "structurally, a unit

approximating to what one could sing in a single breath." (Schoenberg 1967, p.3). We can view musical structure as consisting of a number of hierarchical levels, from the scale of a complete piece, dividing into sections, sections into phrases, and phrases into motifs. Given the primacy of the phrase in tonal melodies, we concentrated on analysis at this level alone.

Elementary textbooks on the theory of music usually identify phrase boundaries with the occurrence of *chord progressions* known as *cadences*. Common examples of cadences which mark phrase boundaries are the 'interrupted' cadence - moving from any chord to 'chord V'¹ and the 'full cadence' which moves from chord V to chord I, often at the end of a piece. It may appear, therefore, that we can analyse phrase boundaries by simply harmonising a melody, then searching the harmonisation for chord progressions - such as V - I, x V, and so on - which indicate cadences and hence phrase boundaries. There are a number of reasons why the problem is not as simple as this. Firstly, given that we know how the potential harmonic units of the melody are to be divided up, assigning a chord to each unit involves knowing which notes are to be harmonised, and which are 'inessential', which may differ widely between musical genres. Secondly, since the rate of change of chords may vary, we have no simple mechanism for isolating chordal units in the first place: the problem is like trying to parse a sequence of letters where we do not know what the lexical units are, and where we have problems in assigning lexical categories even if units are identified. One way to constrain the harmonisation problem is to input major structural boundaries - i.e. cadences marking phrase boundaries. However, since harmonisation requires knowledge of phrase boundaries, we clearly cannot use harmonisation in a simple way to identify phrase boundaries themselves. Even supposing that we could identify the chord progressions which accompany cadences, this would not itself enable us to postulate a *phrase boundary*: the functioning of such chords as cadences also depends on a number of other factors, such as metrical placement of the progression. The structure of phrase groupings is theoretically independent of metrical structure: bar-lines do not necessarily correspond with phrase boundaries. The phenomenon known as 'anacrusis' illustrates this point, where a piece begins with a fraction of a metric unit, and the first bar-line occurs after a number of notes have elapsed.

There are a number of other factors which also seem to relate to boundaries of musical structures, such as repetitions of musical patterns in pitch contour and rhythm. However, such repetitions are often not exact. If we therefore want to say that grouping has something to do with repeated patterns, but not necessarily exact ones, then how inexact can they be, and how do patterns in terms of different musical factors interrelate? A possible response to this problem relates to the fact that the groups containing preferred recurring patterns are often *symmetrical*, or of even size, and so we expect the each phrase to be of the same length. However, this is also commonly not the case - for example, composers often lengthen the final phrase of a piece.

As well as these 'global' factors which relate to musical structure, there is an additional set of

¹For non-musicians, chords are conventionally notated using roman numerals, which refer to the position of their 'root' note, in the scale for a particular key. Capital letters refer to major chords, and lower case to minor chords. For example, chord V is the major chord whose root is the fifth note of the scale of the current key, and chord ii the minor chord on the second degree of the scale.

factors which are more local, and present on the 'musical surface' of the precise pitch, duration and so on of individual notes. Examples of such factors are large interval leaps, rests (not just notated ones, but articulatory ones as well) and changes of some kind, such as those in dynamics or timbre. It seems to be generally the case that phrase boundaries are often additionally emphasised by these local details, but it is not difficult to find counter-examples for them all. So far we have discussed our problem in terms of analysing 'the' structure of a melody. However, in some cases the grouping can be genuinely ambiguous. A different kind of indeterminacy is the case where phrases can overlap with each other, so that the end of one is the beginning of another (the phenomenon of *elision*).

The problem of modelling how a number of interacting musical factors influence our perception of musical grouping structures therefore seems to be one of *specifying how a number of diverse factors interact*. None of these factors appear to be either necessary or sufficient for the analysis of grouping structures. With this knowledge of the general nature of the problem of structural analysis of music, let us proceed to review some of the key areas of the relevant literature. The most comprehensive section is devoted to an analysis of the dominant theory in the field - the "Generative Theory of Tonal Music" of Lerdahl and Jackendoff (1983) - which attempts to incorporate elements of all of the features which we have discussed.

2.2 Existing work relevant to analysis of musical structures

2.2.1 Musical patterns

It is clear that our musical intuitions concerning grouping structure relate in some way to recurring patterns in melodic contour, harmony, rhythm and (possibly) reductional structure. One of the earliest attempts at computational analysis of musical patterns was Simon and Sumner's classic paper "Pattern in Music" (1968). Their approach was to take existing work on inducing patterns in letter sequences and apply this to pattern induction in terms of each separate musical parameter (rhythm, melodic contour and harmonic structure), combining this approach with psychological coding theory. Although their approach is capable of coding a number of pieces when performed by a human analyst, the problem remains of specifying which of the possible codings of the musical sequence is the musically relevant one. Furthermore, given the possibility separate patterns for each of several musical parameters, Simon and Sumner do not address the problem of how these may be integrated.

More recently, this approach has been developed in the form of research on recursion in encoding rules for melodies (Deutsch & Feroe 1981). Some recent research may suggest that symmetry and phenomenal accent may be more important than these rules in encoding melodies (Boltz & Reiss Jones, 1986), but our main criticism concerns combinatorial constraint. Since we are given no description of the mechanisms by which a melody is encoded, we do not know how a system could be developed which could produce 'cognitively economical' encodings without the ability to look ahead to the complete melody. For a melody of only a few bars, a very large number of possible encodings would result, and a combinatorial explosion for any reasonably lengthy melody. Applying pattern-matching techniques to sequences of integers representing intervals or durations produces *combinatorial debris*: without the knowledge possessed by musical

intuition, unguided pattern-matching will fail to select those patterns which human intuition would select as musically relevant. The objections to embracing such combinatorial debris are that they make demands on memory and processing capacity which are psychologically unrealistic, and that it is implausible to suggest that listeners could remember these possible encodings in order to subsequently select the one which finally turned out to be most economical.

2.2.2 Harmonisation and musical grammars

Musical phrases relate to harmonic progressions. We therefore briefly review major approaches to harmonisation, and the development of musical grammars used by some such systems. Music has always been compared with language. The development of Artificial Intelligence techniques for analysing linguistic structures has therefore been accompanied by a relatively small area of research which has aimed to apply these theories to the analysis of musical structures. The music and language comparison must be viewed with some caution, however. For example, the question of determining what 'musical semantics' is, may result from taking this analogy too literally, and many of the sophisticated analytical mechanisms developed for language may turn out to be inapplicable to music. We must therefore look to music itself for a true conception of its structure, and carefully determine to what extent we can use existing techniques of analysis, and to what extent new ones will have to be developed. Winograd's early paper "Linguistics and computer analysis of tonal harmony" (1968) addressed the problem of assigning chord functions to complete tonal pieces. Using a systemic 'feature' grammar, it strove for 'intelligent' combinations of chord functions, resulting from a set of rewrite rules whose terminal nodes were the individual notes themselves. Using a right-to-left parsing algorithm (to eliminate many modulatory possibilities) and a set of heuristics to reduce possible tonalities to be considered, the goal was to find possible completions in terms of the rewrite rules. Since the problem addressed by Winograd is that of assigning harmonic 'labels' to pre-existing chord sequences, it is not directly usable as a method for analysing phrase boundaries in unharmonised melodies. Nevertheless, it was one of the first approaches to demonstrate the feasibility of grammatical approaches to music.

A number of systems which harmonise Bach chorale melodies have been reported recently, including Taft-Thomas' (1985) 'VIVACE' system and Ebcioğlu's (1986a; 1986b) expert system 'CHORAL'. In VIVACE, the parsing procedure used is a combination of bottom-up harmonic analysis, from clusters of notes to chord function, and top-down search for the best paths of chord combinations, working backward from the cadence. What counts as the 'best path' is fundamentally the path which contains the largest number of 4th/5th root movements. In addition, VIVACE works on harmonising only complete phrases. Since our goal is to use a harmonisation to identify those phrase boundaries in the first place, this backward search method would be inapplicable. A number of assumptions concerning harmonic rhythm and treatment of non-harmonic tones are made, which may not be applicable to many other tonal melodies. Firstly, in many tonal melodies harmonic rhythm varies irrespective of tempo, especially in the drive towards a cadence. Since it is the discovery of those cadences which is our goal, we cannot make the simplification of an even harmonic rhythm, and must consider the possibility that at any point in the melody there could be a different chord on each beat.

Secondly, melodic figurations are sufficiently varied in most melodies other than simple chorales for the simple procedure of defining non-harmonic tones as "... any note that is either approached or followed by step and occurs between beats." (Taft- Thomas 1985,p.269) to be inadequate.

The harmonisation problem considered in Ebcioglu's (1986) expert system 'CHORAL', is considerably complicated by an attempt not simply to harmonise melodies in a 'schoolroom style', but in a manner which simulates the *musical style* of J.S. Bach. In harmonising a melody in four parts, a composer will have to simultaneously attend to the 'vertical' harmonic progression of combined voices, as well as to the combination of melodic styles and movements within each voice. These problems were addressed by representing separate 'viewpoints' for 'chord skeletons', chord 'fill in', 'time slices', 'melodic strings', and finally, a viewpoint for a Schenkerian analysis of the harmonisation. The knowledge base of each viewpoint contains *production rules* (approximately 300), such as rules for modulation to a new key, *constraints* for 'early pruning' of possible harmonisations, and *heuristics* which weight desirable properties of solutions in a 'generate and test' method of harmonisation. The program produces creditably correct harmonisations, but ones which "do not look like Bach" (Ebcioglu 1986a). There is no sense in which its control mechanisms and knowledge structures are intended as a *cognitive model* of how listeners infer harmony. The program does succeed in harmonising simple melodies using its extensive knowledge base, and is important in demonstrating the role of powerful heuristics and *multiple viewpoints* in music cognition.

Work on harmonisation commonly uses the notion of grammar to formalise the regularities observed in chord progressions for specific musical genres. Developing effective chord grammars for given musical styles is, however, a considerable research problem. A good example of such research is Steedman's (1984) grammar for the 'twelve bar blues'. Examples of Steedman's rules are the top-level rule which acts as 'input' to the others

Rule0: S12 → I I7 IV I V7

and the rule

Rule 2: x(7) → x(7) subdominant(x)

which states that a chord or its seventh can be rewritten within the same time-span to that chord or its seventh, and the subdominant with respect to that chord. Although Steedman's aim was to characterise a limited musical style, he speculates that the rules derived for his limited domain are

"... part of a larger set of rules which characterise a much more comprehensive set of chord sequences that are similarly musically coherent within the tradition of Western Tonal Harmony."

(Steedman 1984)

One further possible approach to harmonisation should be mentioned, derived from the subfield of cognitive sciences called connectionism. The approach has been used to model low-

level aspects of perception, such as vision (see Hinton & Anderson 1981), but may be less appropriate for higher level cognitive processes. Musical applications of connectionism include Bharuca's (1987) model of musical harmony, and Lischka's (1987) model for harmonisation of Bach chorales. We would maintain that for modelling high-level cognition ('chunking') in the auditory domain, the connectionist approach is at the wrong level of explanatory description², but that it may be suited to lower levels of harmonic processing. Since our ultimate goal is to use a cognitive model of musical structures as a domain model of a tutorial dialogue model, it is clear that we need a high level symbolic model which forms a suitable basis for tutorial communication.

From our selective review of important research in computational harmonisation, we can see that most existing work does not address the problem of how harmonisations are unconsciously inferred for unharmonised melodies, in the absence of a prior knowledge of the placement of musical structural boundaries which define segments to be harmonised.

2.2.3 Lerdahl and Jackendoff's "Generative Theory of Tonal Music"

The publication of "A Generative Theory of Tonal Music" by Lerdahl and Jackendoff (1983)³ was pioneering in the field of music and cognitive science, and most subsequent work is based on it to some extent, including our theory presented in this chapter. Despite the title of their work, the psycholinguistic component of the theory lies rather in theories of prosodic structure (relative stress and accent in pronunciation) and Gestalt theories of grouping applied in the auditory domain. The theory attempts to provide a formal specification of the intuitive rules by which a hypothetical ideal listener, educated in the Western tonal tradition, imposes intelligible structure on auditory musical input. There four aspects of musical intuition which the theory attempts to capture - (i) grouping structure; (ii) metric structure; (iii) time-span reduction; and (iv) prolongational reduction:

"...*grouping structure* expresses a hierarchical segmentation of the piece into motives, phrases and sections. *Metrical structure* expresses the intuition that the events of the piece are related to a regular alternation of strong and weak beats at a number of hierarchical levels. *Time-span reduction* assigns to the pitches of a piece a hierarchy of "structural importance" with respect to their position in grouping and metrical structure. *Prolongational reduction* assigns to pitches a hierarchy that expresses harmonic and melodic tension and relaxation, continuity and progression."

(pp. 8 - 9.)

Within each of these sections, rules are broadly divided into two groups:

(1) *well-formedness ('wf') rules*, which specify possible structural descriptions⁴, and

² The issue is a highly contentious one. See Broadbent (1985), and replies in Rumelhart & McClelland (1985) on the issue of connectionism and reductionism concerning 'levels' of explanation and processing.

³ All future references to Lerdahl and Jackendoff in this thesis are to the 1983 version of their work.

⁴ In addition, there is a small number of other rules termed *transformational rules*, which operate on hierarchical structures provided by wf rules to account for phenomena such as elisions and

(2) *preference rules* ('pr's'), which identify which of the possible well-formed analyses correspond to experienced listeners' hearings of the piece.

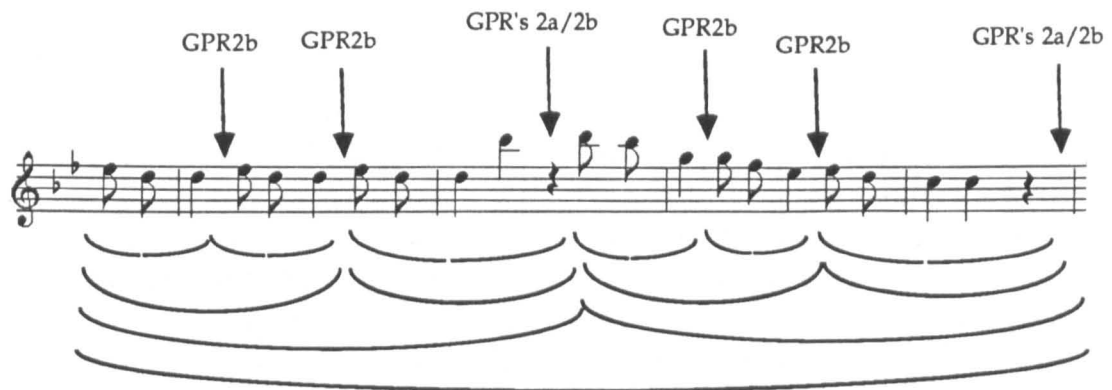
For our purposes here we shall only discuss grouping structure and time-span reductions in any detail.

2.2.3.1 Grouping structure

Grouping well-formedness ("wf") rules establish that a piece is a group, that a piece must be exhaustively partitioned, that groups may not overlap, and so on. As with vision, perception of musical grouping is argued to be a matter of *relative strength*. Figure 2.0 shows a short example of a possible grouping analysis ('grouping preference rule' is abbreviated to 'GPR'), as analysed by Lerdahl and Jackendoff (p.48) with the grouping preference rules generating the boundaries marked.

Figure 2.0

Example grouping analysis of theme from Mozart's G minor Symphony



Reading from left to right, the marked boundaries are predicted by surface features in the music, as follows:

GPR2b - differences in attack points, and

GPR2a- time intervals, or rests.

It is clear to intuition, that the boundaries marked are correct. However, many more boundaries are predicted by the rules than those shown⁵, which leads to the problems of *conflict resolution*, *rule reinforcement*, and *interaction of global and local factors*, for which their theory provides no clear algorithmic solution. We can also see from this example that grouping is strongly influenced by symmetry of groups and recurring patterns, or *parallelism*. The problem, from the point of view of formally specifying parallelism, is to limit the space of possible ways in which a melodic fragment may be varied and yet remain 'similar', since it is clear that otherwise any section could be transformed into any other. The concept of overlapping phrase structures.

⁵ We have performed a simple computer implementation of a number of the rules, and found many additional boundaries generated by their strictly formal application, than those predicted by Lerdahl and Jackendoff's hand-done analyses (for example, grouping preference rule GPR3a).

parallelism is invoked throughout each section of the rules and may thus be viewed as essential to it. The lack of a set of formal rules for parallelism therefore represents an important problem for the theory.

It is clear that although the part of the theory of Lerdahl and Jackendoff concerned with grouping structure identifies many of the musical factors involved in grouping perception, the theory does not provide an algorithm for grouping analysis, nor a cognitive architecture within which to situate such an algorithm.

2.2.3.2 Time-span reductions

The idea of reduction is designed to capture such facts of musical listening as recognition that variations are like each other and that some passages sound like ornamented or else simplified versions of others. The following 'Reduction Hypothesis' is central to the theory of Lerdahl and Jackendoff :

"The listener attempts to organize all the pitch-events of a piece into a single coherent structure, such that they are heard in a hierarchy of relative importance."
(p.106)

In order to establish criteria for determining this hierarchy of importance, Lerdahl and Jackendoff appeal to the hypothetical listener's intuitions concerning the effect produced by removing certain pitches, and the notion that one set of pitches can be viewed as an 'elaboration' of another. For example, the total effect of reducing out the cadence from a tonal piece would be much more dramatic than reducing out an ornamental appoggiatura. Criteria of 'pitch stability' or elaboration - upon which the relative structural importance of pitch-events are based - are derived from traditional concepts in music theory, such as relative closeness on the cycle of fifths, closeness to root position for chord inversions, and the notion of rhythmically and harmonically 'inessential' notes. Clearly, the theory requires further criteria to determine the musical segments within which groups of notes are compared with others. A formal specification of the domains of analysis based on the grouping and metric part of their theory, which they term *time-spans*. Broadly speaking, the analytic process involves identifying the time-spans, identifying structurally important events within the time spans and constructing trees which express the relation 'is-an-elaboration-of', working essentially 'bottom up', applying principles for identifying the head of a time-span recursively. Given this definition of possible time-span reductions, the time-span reduction preference rules determine how a head should be chosen from the heads of the immediately contained time-spans.

We have discussed the reductional component of Lerdahl and Jackendoff's theory because we argue as part of our own model that the relationship between reductional and grouping structure is an important one in grouping perception, for the following reasons:

- musical parallelisms may be more apparent at deep reductional levels,

- grouping structure and reductions interrelate since reductions require musical segments to be predefined over which a reduction can take place
- time-span reductions require an abstract fundamental underlying structure called a "normal form", which relates directly to large-scale phrase structures.

Lerdahl and Jackendoff state that reductions and grouping must interrelate in some way - GPR7 states "Prefer a grouping structure that results in more stable time-span and/or prolongational reductions" (p.52) - but they do not enlarge upon this statement.

2.2.3.3 Conclusions concerning Lerdahl & Jackendoff's theory

The importance of Lerdahl and Jackendoff's work lies in the extent to which it has stimulated a large amount of research in music and the cognitive sciences. Any problems identified do little to lessen the importance and value of their work. Let us finally attempt to make some general hypotheses as to why the theory of Lerdahl and Jackendoff (1983) - which embodies intuitively correct musical insights - is theoretically unworkable and inherently unalgorithmic. There seem to be two principal reasons, and two major problems which remain. The reasons are as follows:

(1) *the theory is essentially 'flat'*

The theory is essentially 'flat' in that a single rule-based formalism for analysis of musical structures is proposed, with no distinction between low-level perceptual phenomena and higher level cognitive processes. Theories in other areas of cognitive science, such as speech recognition and vision, usually incorporate a conception of successive *levels* of processing - in the case of speech, possible word boundaries are detected in the acoustical speech input, and the speech is parsed by a combination of these low-level features and high-level syntactic, semantic and pragmatic constraints. In Lerdahl and Jackendoff's theory, therefore, the features treated by the grouping preference rules, for example, should be precisely quantified at the *acoustical* level, and integrated into a model for higher level harmonic and reductional processing. Furthermore, successive levels of processing must operate for a number of concurrent processes - perception of grouping, metre, patterns, reductions, tension-and-relaxation - and we are given no model for how these component processes *interact*.

(2) *the theory is essentially static*

There is little sense in which the theory provides us with a model for the underlying cognitive *processes* for music cognition⁶. Where some prototheory can be elicited from the overall research, it is clear that the direction and amount of processing required is not at all psychologically plausible. For example, a listener cannot remember all possible grouping boundaries in a melody, then subsequently apply metarules to resolve conflicts, nor simultaneously conceive of complete reductional parse trees, and prune alternatives. We suggest

⁶ Since the publication of their book in 1983, Lerdahl and Jackendoff have begun to address some of these problems of specifying underlying psychological processes. Details of an initial approach have been specified in unpublished manuscripts gained by the author's personal communication with Lerdahl in Summer 1988. Our criticisms here have referred exclusively to the theory as it is known to the general cognitive science and music research community.

that this problem stems from a fundamental ambiguity in the theory, between the goals of a theory of music *cognition* or *perception*, and the *explicit analytical processes* performed by a human musicologist when looking at a musical score.

Two major problems which remain are

- (1) *how is parallelism to be formalised?*
- (2) *how can we produce a model of how the component processes interact?*
(i.e. grouping, metre, harmony, reductions).

The model for analysis of grouping structures described in section 2.3 of this chapter attempts to address precisely these two problems. In the remainder of this section we shall briefly review some of the work which was based on Lerdahl and Jackendoff's research.

2.2.4 Approaches to refining the theory of Lerdahl and Jackendoff

The approach of assigning numerical 'strength' weightings to 'preference' rules which predict grouping boundaries has been adopted in various forms by a number of workers (Tenney & Polanski, 1980; Heefer & Leman, 1986), in accordance with Lerdahl and Jackendoff's statement that grouping perception is a question of relative strength. In Tenney and Polanski's work this takes the form of deriving a 'distance metric' to relate strengths of rules which initiate possible group boundaries ("clang initiation"). Heefer & Leman (1986) have implemented seven of Lerdahl and Jackendoff's Gestalt rules for "primary chunking", using 'segmentation demons' which scan the musical surface for potential boundaries and store hypotheses in a blackboard-like data structure. One of the major problems which this approach faces is the resolution of conflicts between alternative hypotheses concerning musical grouping boundaries in a musically relevant way. As they point out, after reviewing the psychological studies of Deliege (1987) in this field,

"... resolving conflicts between hypotheses (concerning musical grouping boundaries) is a problem for which there exists no single strategy."

(Heefer & Leman 1986, p.83)

We have little assurance that adopting their method does not in effect amount to Tenney and Polanski's problem of having to assign different weights to parameters for each piece under consideration⁷. A similar approach to using numerical weightings is used in the production system approaches to modelling musical listening of Ashley (1988) and Jones, Scarborough and Miller (1988), which use a 'blackboard' database (Erman, Hayes-Roth, Lesser & Reddy, 1980) to coordinate a set of production systems for the metrical and grouping analysis components of Lerdahl and Jackendoff's (1983) theory, together with a component for analysis of tonality.

Lerdahl and Jackendoff (1983) themselves eschew assigning numerical weightings to their preference rules which mark possible group boundaries, citing Winston's (1977) recognition that a resort to numerical weightings in a program designed to learn structural descriptions from

⁷ They report analysing the same musical example, Debussy's *Syrinx*.

examples was mere computational expediency, which did not further our understanding. There is a more fundamental reason why the assignment of numerical weightings would be inappropriate for their theory: since the musical parameters referred to in the rules such as dynamics and articulation are not themselves precisely quantified (this would need to be done at the *acoustic* level), the erection of a system of numerically based decision procedures on this foundation would be a kind of spurious accuracy.

2.2.5 Summary and prospects for a cognitive theory of musical structure

Research in Music and the Cognitive Sciences is currently at a very early stage of development in comparison with other areas of Artificial Intelligence and Cognitive Science. Some indication of the state of the field in 1989 is indicated by the fact that the first symposium devoted exclusively to Music and the Cognitive Sciences was held at IRCAM in Paris in Spring 1988, and the First Workshop in Artificial Intelligence and Music was held in Summer of the same year at the AAAI conference in St. Paul Minnesota. From our review in this chapter we can see a picture of the field as largely fragmented, with approaches which are based on AI techniques such as expert systems (which thus make no claims to cognitive validity) and other approaches based on adaptations of natural language parsing techniques beginning to be developed for very limited domains. Most subsequent research has been therefore based on an attempt to take the Lerdahl and Jackendoff rules, and place them into some kind of production system. These approaches fail to address the fundamental problems with the theory, since a single rule-based formalism would require a deeper knowledge representation to do justice to successive levels of musical processing, which gives us a 'transparent' cognitive architecture for specifying how component processes *interrelate* - a major motivation of our own model presented in the next section.

2.3 Towards a Cognitive Model for Musical Phrase Structures

2.3.1 Introduction

From our review of artificial intelligence approaches to the analysis of musical structures we concluded that the best prospect for developing a formal cognitive model⁸ for analysis of musical phrase structures ("grouping structure") was to recast the musically correct insights of the Lerdahl and Jackendoff theory within a cognitive architecture which integrates multiple processes for harmony, metre, grouping and reductions, and which contains a conception of levels of processing, from low-level features on the musical 'surface' to higher level harmonic and schematic processing. The model which we propose for addressing this research problem is called 'GRAF' (GRouping Analysis with Frames), and is based on a reformulation of the Lerdahl and Jackendoff theory. GRAF was developed in two stages:

- (1) development of a musical harmonic 'syntactic' parser ('AGA', Automated Grouping Analysis system),
- (2) development of GRAF itself, which integrated AGA.

⁸ Since this is a thesis in Cognitive Science applied to Intelligent Tutoring Systems, we assume that readers are familiar with the goals and methods of computational models in this field, in terms of producing fully formalised and testable theories. See Cohen (1983) for a full discussion of these issues.

Accordingly, we describe it in two parts, beginning with the 'syntactic' parser for musical chord functions, and a critique of the psychological plausibility of this approach, followed by a description and critique of GRAF. We conclude with prospects for further research to develop a cognitive model for grouping analysis within an overall model of musical cognition, and a description of how GRAF could be used as a domain model in the dialogue model described in the remaining chapters of this thesis.

2.3.2 Overview of AGA and GRAF

AGA can be viewed as a combination of syntactic processing of harmony with the 'diversification' approach to resolving uncertainty (Cohen 1985). The approach involves specifying a set of decision criteria for ranking a set of possible solutions. In AGA, the set of possible solutions is the set of well-formed syntactic parses according to a context-free grammar of chord function, which identify possible grouping boundaries at the level of the phrase. The principal decision criteria used are

- (1) to rank the possibilities according to the maximal extent of matching in melodic contour between phrase groups, at the highest reductional level, and
- (2) to limit the parse tree construction to strings of chords which predict phrase boundaries at positions which are "reinforced" in terms of local details, according to the Lerdahl and Jackendoff grouping preference rules, described earlier in this chapter.

We can thus see how AGA builds on the work of Lerdahl and Jackendoff and incorporates the major features of their theory within a new model. An additional research contribution of AGA is its incorporation of the beginnings of a *formal theory of musical reductions* based on the hierarchical relationships incorporated in harmonic parse trees, and the idea that musical parallelisms will be more evident at reductional levels rather than on the musical surface.

GRAF was developed in response to a number of problems of psychological plausibility with AGA, in terms of the amount and direction of processing, and results in experimental psychology of music (McAdams 1987; Stoffer 1985) which suggested that high level *schemata* could explain a number of features of grouping perception. Grouping schemata can be viewed as a more psychologically plausible theoretical representation of the 'well-formedness' and 'symmetry' grouping rules of Lerdahl and Jackendoff, and the Lerdahl and Jackendoff surface grouping preference rules are now recast as low-level *feature detectors*, which generate possible matchings with grouping frames. The AGA harmonic parser is incorporated as part of the frame-matching process in the sense that it can be used as a *discrimination test* by limiting harmonic processing to the problem of determining whether a putative grouping boundary defines a section of the melody which could function as a phrase boundary, in terms of its harmonic structure.

2.4 AGA: a syntactic harmonic processing approach

2.4.1 The aim of the model

Grouping is arguably the most important level of musical understanding: it performs the function of extending the scope of the piece of music which can be available for processing at any given

time by 'chunking' pitches into higher level units. Modelling how listeners 'understand' melodies is restricted here to the problem of how listeners assign grouping structures to *tonal melodies* by processing discrete serially input pitches, in combination with their knowledge of other pieces. Since we take the view that any approach to modelling grouping perception must do justice to how a number of processes interact, studying melodies forces us to specify these interacting processes - such as the unconscious inference of harmony - in conjunction with grouping perception. Furthermore, we agree with Lerdahl and Jackendoff (1983) that separate grouping analyses are probably required for each polyphonic line, so the study of grouping in single melodic lines is a necessary precursor to this wider problem.

The Automated Grouping Analysis system ('AGA', Baker 1989b) is a computational model of how listeners process discrete pitches to chunk tonal melodies at the level of the phrase. It has been partially implemented in Common Lisp and the object-oriented extension 'flavors' on an Apollo Domain AI workstation, interfaced to a Yamaha KX88 touch-sensitive keyboard. The system starts from a sequence of discrete pitches represented symbolically in Lisp, and so is probably best understood as modelling processes at the level of what McAdams terms "the extraction of the musical lexicon" (McAdams, 1987), at a level of abstraction above Lerdahl and Jackendoff's (1983) view of the "musical surface" as "the physical signal of a piece when it is played". Most of our research effort was applied to reformulating and formalising the Lerdahl and Jackendoff theory, in combination with other research in the cognitive science of music, to define a theory which was inherently *algorithmic*. For this reason, computer programming was used only in the service of testing selected parts of the theory which we developed (embodied in AGA). For example, we tested the extent to which the grouping preference rules of Lerdahl and Jackendoff could serve as an algorithm for grouping analysis by implementing them as a production system, and we explored the role of "parallelism" in music perception by implementing a pattern-matching system. In addition, we tested the algorithmic nature of the chord grammar proposed in AGA by implementing a chart parser, together with a musical reductions program which we describe later. Since these program components had fulfilled their purpose in testing alternative possible approaches (of which AGA was the result) implementation and integration of these components was not carried to completion. For this reason, we do not describe actual program output, but rather a set of algorithms developed from exploratory programming used in the service of theory development.

Criticisms of the work of Lerdahl and Jackendoff from the perspectives of psychology and music theory are now well documented in the literature (Peel & Slawson, 1984; Clarke, 1986). Possibly the most serious problem facing the grouping component of their theory concerns the notion of 'parallelism' (Lerdahl & Jackendoff, 1983, p. 52 ff). As Lerdahl and Jackendoff themselves state, the problem is not only to formalise what parallelism is (how similar must two passages be before they are construed as parallel?), but of how to compare parallelism with anything else. The problem is an instance of a more general problem with the theory, of formalising how global features interact with local surface detail. Although we recognise Lerdahl and Jackendoff's important theoretical advance in identifying grouping structure as an analytically separable component, we maintain that *any approach to grouping analysis must concentrate on*

the problem of how grouping interacts with other musical factors.

AGA exploits a limited set of these interrelationships⁹, being the relationship between *grouping structure*, *parallelism* and *time-span reductions*. The relationship obtains in two main ways: (1) between grouping at the phrase level and the 'normal forms' of time-span reduction; (2) between grouping at the phrase level and the deep linear parallelism between antecedent and consequent phrases at successive levels in time-span reductions. Relationship (1) is depicted in Figure 2.1, showing the normal form for the so-called interrupted form, which identifies a kind of template or schema to which time-span reductions must conform (Lerdahl & Jackendoff refer to it as a "background" structure, or "skeleton"; 1983 p.139 ff). Since time-span reductions for a piece must conform to a normal form for that piece, an acceptable time-span reduction will depend on the normal form being based on a correct identification of *phrase boundaries*. Once a set of grammatically well-formed time-span reduction trees has been generated, each of which embodies hypothesised phrase boundaries, we can use relationship (2) to constrain the space of possible parse trees to those which accord with musical intuition.

Figure 2.1

**The 'normal form' for the interrupted form,
(redrawn from Lerdahl & Jackendoff (1983), p. 141).**

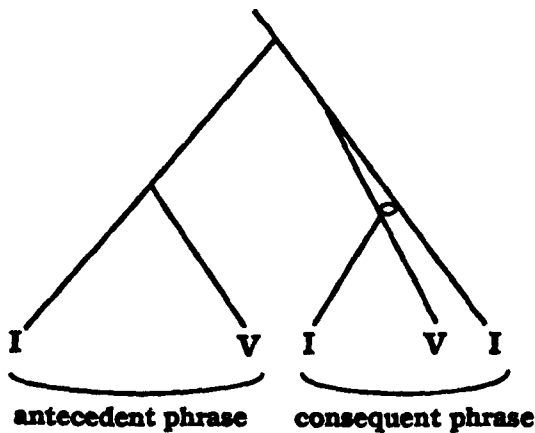
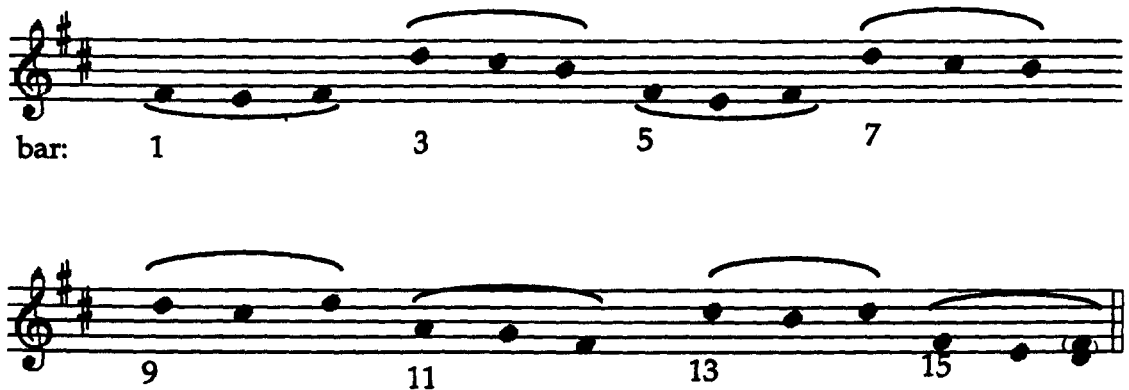


Figure 2.2 shows an example of deep reductional parallelism exhibited by the J.S.Bach chorale "O Haupt voll Blut und Wunden" (St. Matthew Passion), according to the Lerdahl & Jackendoff time-span reduction (Lerdahl & Jackendoff 1983, p.144).

⁹ We assume that key and metre are already determined, as inputs to the model.

Figure 2.2

A section of the time-span reduction from the chorale "O Haupt voll Blut und Wunden", (St. Matthew Passion), J.S.Bach, (redrawn from Lerdahl & R.Jackendoff,1983).



Our underlying hypothesis is that

for certain well-defined musical subcultures, phrase level groups identified by parsing according to the most appropriate normal form will exhibit a greater degree of parallelism at some reductional level than any other phrase level groups which are grammatically well-formed.

"Greater degree" of parallelism is defined in the next section, and the term "musical subculture" is used in the sense understood by Steedman (1984), to be a subset of tonal pieces which can be characterised by a formal context-free grammar.

The normal form of Figure 2.1 is notationally equivalent to the rewrite rules

$s \rightarrow L R$

$L \rightarrow I V$

$R \rightarrow I V I$

where s is the melody, L is what we term an 'antecedent phrase', R is a 'consequent phrase', and I and V are chord function symbols with their usual theoretical meaning. It is possible to abstract from Lerdahl and Jackendoff's time-span reduction theory (in combination with conventional music theory) a more extended grammar which encapsulates chord function:

$ii \rightarrow vi ii$, $vi \rightarrow iii vi$, $V \rightarrow ii V$, are examples of rules which in effect extend a left-branching cadence backwards. $I \rightarrow I V$, $I \rightarrow I IV$, are examples where a "structural beginning" (Lerdahl and Jackendoff) is extended forwards. Any number of 'trivial' substitution rules are possible, according to the subculture being modelled, for example $I \rightarrow vi$, $V \rightarrow V7$, $V \rightarrow vii7$, and so on. Finally, the modelling of different musical forms requires different normal forms, or different combinations of normal forms as the highest level rule of the grammar. This represents a considerable limitation of the model, as we discuss later in this

chapter.

2.4.2 The Automated Grouping Analysis System

AGA can be divided into four main phases: lexical analysis (harmonisation), parsing, reductions, and pattern matching under limited transformations. The melody, its metre and a chord lexicon are input to the 'lexical analyser', which returns a set of chord functions assigned to each metric unit, from the bar level down to each beat. This is taken as input to a *chart parser*, together with the harmonic grammar appropriate to that melody, which returns a set of possible parse trees for the melody. Each parse tree (together with the chord lexicon) provides the basis for time-span reductions, each of which are evaluated for maximal parallelism displayed at each level. Finally, a simple algorithm is used to assess the degree of parallelism displayed, to sort the small number of possible boundary-sets into order of preference. The final outcome is a small ordered list of possible phrase-level boundary sets for the given melody.

2.4.2.1 From the musical surface to the musical lexicon

The 'lexical analysis' program generates a set of lexical items (chord functions) for metric units in the input melody. Melodies are represented as lists of pitch events of the form

(<note-number n1> <start-time n1> <velocity n1> <duration n1>)

Duration is measured in units of the shortest note duration, and start-times are expressed as the sum of durations. Pitches are represented as numbers of semitones above or below a defined keynote (= 0) for the melody. A melody is simply a list of these pitch-event lists, being a representation which facilitates the use of elegant and efficient recursive list processing algorithms. Since harmonic units are defined by metric rather than grouping divisions, the program needs information as to how the list of notes should be divided into bars and beats, including the position of the first bar. In addition, it needs a 'chord lexicon', being simply a list which pairs chord functions with the notes which can match that function. For example, the first element of the list

(I.(0 4 7) ii.(2 5 9) ii7.(0 2 5 9))

states that where 0 is defined as the keynote, chord I matches with any group of tones which are identical to, or the octave equivalents of, that keynote, and the tones four and seven semitones above it, respectively. All of these kinds of information are stored in a special data structure (an 'object') which is created for that melody.

Lexical analysis can be divided into two main phases: (1) isolating metric units in the melody (2) returning the possible harmonisations for metric units, according to the chord lexicon. At the most general level, the program attempts to return a harmonisation for each bar of the melody, and within each bar for the half-bars and each beat (depending upon the metre of the melody). In this respect the harmonisation process differs from existing work in this field, since we cannot make assumptions that the piece will display a particular harmonic rhythm (Taft-Thomas 1985), nor do we possess an existing harmonisation, to which we need to assign a harmonic analysis (Winograd 1968). Since metric and grouping boundaries are distinct, we cannot assume

that the phrase boundary will coincide with the bar or half-bar lines.

For each metric unit, the 'harmonise' function is called under a series of *melodic viewpoints*. Melodic viewpoints correspond to the possible kinds of melodic figuration for that melody, and are implemented as functions which aim to isolate the pitches in a metric group which should be treated as harmonic tones. Once isolated, the harmonic tones (when compressed to equivalents within a single octave) are simply matched with the chord lexicon. The harmonisation for a metric unit is the combination of matched chord functions, under each applicable melodic viewpoint. At present, melodic viewpoints implemented are 'onbeat-notes', 'onbeat-passing-tones', arpeggio, 'upper-auxiliary', 'lower-auxiliary' and 'changing-notes'. In effect, they correspond to possible harmonic reductions for that section of the melody. For example, the 'onbeat-notes' function considers the possible melodic figuration that the harmony notes are those on the beat, and so simply returns these notes to be matched with the chord lexicon. The 'arpeggio' viewpoint attempts to treat all of the notes as belonging to the same chord: if it succeeds, then no further viewpoints are considered for that unit, nor for the metric units into which it could be divided (the melodic section is already 'reduced' to tones which can all be viewed as part of the same chord, this being the goal of alternative melodic viewpoints).

It is clear that music is so potentially 'lexically ambiguous' in comparison with natural languages, that a great deal of constraining knowledge, or heuristic guidance, is required to limit this ambiguity. In formal terms, our model shows that the problem of inferring harmony implicit in a group of pitches, in a purely 'bottom-up' manner, is a *combinatorially unconstrained* problem. Our notion of melodic viewpoints is designed to capture the set of possible melodic figurations for groups of melodies, from which we can attempt to infer harmonic tones, but for any set of pitches considered devoid of musical context it is clear that any number of - or even all - viewpoints could be assigned, which gives us very little theoretical leverage. Take, for example, the first three notes of the second bar of Figure 2.3, and consider possible harmonisations using a single chord., where (at least) the following melodic viewpoints apply:

onbeat tones: select tones on the beat as harmonic tones;

passing notes: for any three consecutive three tones which move in an ascending or descending diatonic scale, select the first and third as harmonic tones.

Figure 2.3 Harmonisation of first two bars of Mozart piano sonata K545.



[using a restricted chord lexicon with only I,IV,vi,ii,V.]

value returned by the function call

```
> (lexical-analysis "mozart-k545" "chord-lexicon" '((time-sig 4 4) (beatdivn 4)) 1 1)
(((bar 1) (full-bar (I)) (first-half-bar (I IV vi)) (second-half-bar (I))
  (beat-1 (I IV vi)) (beat-2 (I IV vi)) (beat-3 (I vi iii)) (beat 4 (I V)))
 ((bar 2) (full-bar (I)) (first-half-bar (V)) (second-half-bar (I vi))
  (beat-1 (V)) (beat-2 (I V)) (beat-3 (I vi)) (beat 4 (I vi))))
```

With the first viewpoint, the single b natural as the harmonic tone would return possible harmonisations iii, V and vii, given a suitable chord 'lexicon'. For the second viewpoint we have a b natural and a d natural to harmonise, which is harmonised as chord V or vii, so the total possible harmonisations are iii, V and vii. Add to this the fact that many genres would require a much richer harmonic vocabulary, and that harmonic rhythm could vary from piece to piece, and we see that considered 'bottom up', tonal music possesses a very high degree of indeterminacy at the level of individual harmonies (in comparison to lexical categories which could be assigned to isolated words, for example). The fact that listeners simply could not - in information processing terms - unconsciously infer harmony with this degree of indeterminacy reveals the extent to which listeners must be guided by *top-down* expectations concerning likely 'grammatical' continuations from one chord to the next, and by knowledge of harmonic rhythm and vocabulary. A simple method of adjusting the model to incorporate the listener's use of top-down knowledge would be to integrate the harmonisation and parsing processes, to make top-down predictions for chords to continue a sequence, and to incorporate genre-specific knowledge in the form of *heuristics*. Using such heuristic knowledge simply means that search is constrained by knowledge of the genre concerned, which a listener might be expected to have acquired. Specifically, the program needs knowledge of the following kinds: (a) of which chords should be allowed in the lexicon; (b) of harmonic rhythm (it is only at the drive to a final cadence, for example, that the harmonic rhythm usually extends down to the beat level); (c) of which kinds of melodic figuration occur in that subculture. Finding such a set of heuristics which adequately constrain a particular 'musical subculture' is itself a non-trivial problem, especially since heuristic knowledge of harmonic rhythm may vary from piece to piece, and within particular pieces.

2.4.2.2 The parsing process

A *chart parser* is used to build possible parse trees from the lexical items according to the grammar. We briefly describe the parsing process not because such parsers are not well known

(Winograd 1983, p. 116), but in order to describe our specific approach to using this method.

Prior to parsing, the output of the lexical analyser (chords assigned to metric units) is initialised into a 'chart' data-structure. The chart consists of *edges*, spanning *vertices*, which for our purposes can be thought of as the time-spans which chord functions apply to, and the points between the edges, respectively. For example, in Figure 2.3 an 'edge' would be stored for each of chord I, chord IV and chord vi, assigned to the first beat. These edges would be recorded as 'outgoing' edges to the first vertex (v0) and incoming edges to the second vertex (v1).

During the course of the parse, new edges are built by continuing existing edges according to the rules in the grammar, and are stored in the chart as incoming or outgoing edges to a vertex. For example, in Figure 2.3, if we have an edge from v0 to v1 with label I, and an edge from v1 to v2, with label IV, then from a rule $I \rightarrow I\ IV$, matching with the right-hand-side enables us to build a new edge from v0 to v2, with its label as the left-hand-side of the rule, i.e. I, which would be placed on the chart (providing such an edge was not already present, which it is in this example). The parser terminates, for our purposes, when there are no further edges which could be continued, and returns all the edges which span the whole chart.

In order to use information residing in the chart for the reductional process, each new edge needs to record its 'children', i.e. the edges from which it is built. In order to retrieve the full parse tree you simply have to recursively look for the children of an edge in the chart, terminating with edges which have no children.

The main reason for using a chart parser here is that it provides a completely flexible data-structure with which to conduct the parsing process in a manner which best models human processing. The main possibilities are to parse bottom-up, or top-down, and either left-to-right, right-to-left or some complex combination of these methods (see Elsom-Cook & du Boulay, 1986). At present, AGA uses a combination of bottom-up and top-down methods (a 'left corner parse', Johnson-Laird 1983), which is arguably the most psychologically plausible mechanism for natural language processing (Johnson-Laird, 1983, p.355). Formal properties of the grammar (as it stands) which we use would preclude a purely top-down approach in any case. For any top-down parser, a grammar which is left-recursive will loop indefinitely, since the first element of the right-hand side will be continually proposed for extension at the same vertex. Left-recursive rules have the form $x \rightarrow x\ y$ (for example, our rule $I \rightarrow I\ V$). Parsing backwards will not avoid this in our grammar, since we also have right-recursive rules, such as $V \rightarrow ii\ V$. Top-down parsers *can* be used with such a grammar, but only if it is rewritten to a *weakly equivalent* form prior to parsing (Winograd, 1983, p. 112), where, for example, the rules $A \rightarrow I\ V$, and $I \rightarrow I\ V$ would be replaced in the grammar by $A \rightarrow I\ x\ V$ and $x \rightarrow V$. The two grammars are weakly equivalent in that they will recognise the same set of strings in the language, but will assign different parse trees. We have not adopted this approach since the insertion of such symbols is clearly psychologically arbitrary, and would greatly complicate the reductional procedure based upon such parse trees.

The parsing process makes extremely high information processing and storage demands, since many edges will be created, stored and used to continue other edges, which do not ultimately lead to a parse for the full melody. It seems unlikely that human information processing would be so inefficient and still be able to parse serially input pitches in real-time, given limitations on working memory. The parsing process can be viewed as a *search* through the space of well-formed substrings in the chart, with a set of *heuristics* to guide that search. One source of heuristic rules - there may well be others - can be found in the Lerdahl and Jackendoff grouping preference rules. We have implemented four of these rules (GPR2a, GPR2b, GPR3a, GPR3d) as functions which return a set of positions in the melody where each rule is instantiated. It seems to be generally true that although there are many redundant boundaries, nevertheless, the musically acceptable phrase boundaries are contained within the total set predicted. Accordingly, we can apply these rules as a rather weak set of heuristics, by *not* building a new edge where its starting and end vertices predict a possible phrase boundary which is not a member of the set predicted by the grouping preference rules. Heuristic search methods for solving problems are characterised by the fact that they are not *guaranteed* to find a correct solution. However, Ebcioğlu (1986b) has shown convincingly, that the specification of a detailed set of appropriate heuristics can be an effective method for modelling a specific musical style. Our use of grouping preference rules as a set of *informal heuristics* significantly differs from their role in Lerdahl and Jackendoff's (1983) theoretical system. In our model the theoretical role of grouping preference rules is as a set of "clues" (Deliege 1987) present on the musical surface, which guide harmonic parsing. Finally, it must again be stressed that our model is strictly an idealisation which assumes complete and exhaustive perception of possible harmonisations. Clearly, our perception of harmony is much more fragmentary, and there will be many interpersonal differences in the extent to which harmonies and their interrelationships are perceived.

2.4.2.3 The reductional process

The reductional procedure is based on the parse trees which can be extracted from the chart data structure, and a number of the Lerdahl and Jackendoff principles for performing time-span reductions. We assume that our grammar has correctly encapsulated the way in which the notes within the time-span of one harmony relate to an adjacent time-span, since the grammar specifies how the chords *function*. For example, if the adjacent harmonies I and V have the chord I as a parent, then according to the rule $I \rightarrow I V$, the notes which originally fell under the compass of chord I will be retained in the reduction in favour of those covered by chord V, since the rule expresses the fact that I 'dominates' V in this context. Where both siblings in a tree match the parent chord - such as in the rule $I \rightarrow I I$ - then the context of the subtree in either an antecedent or consequent group is used to ascertain the 'dominance' relation: in antecedent groups, the left-hand sibling dominates, and the converse is true with consequent groups. The resulting reductions are primarily harmonic in nature - they reveal a kind of 'harmonic skeleton' for the melody. Most reductional procedures seem to be a combination of the top-down approach of looking for larger scale recurring linear patterns with a bottom-up reducing away of 'inessential' notes. Our reductional algorithm takes this into account by exploiting the fact that the parsing process has already identified the larger scale top-down

divisions in the melody. The reductional algorithm therefore combines the exploitation of hierarchical harmonic relationships expressed by the parse trees with a bottom-up reduction of non-harmonic tones, and tones which are less 'harmonically stable', using a number of simple criteria specified by Lerdahl and Jackendoff (1983, p. 161 ff). Figure 2.4 summarises the reductional process.

Figure 2.4

A simplified algorithm for performing musical reductions

```

function:  reduce
inputs:    melody    - the melody
           tree       - the parse tree for the melody, derived from the chart
           lexicon    - an association list of chord functions with lists of notes
           normal-form - the highest level rule for the grammar

algorithm :
  level-0 ← list of pairs: (<group of notes><terminal chord symbol>)
  level-1 ← remove-non-harmonic-tones level-0
  level-2 ← collapse-repeated-tones level-1
  level-3 ← remove-unstable-tones level-2
  successive-levels
  do
    for each sibling pair in level-3,
      retain the single tone whose chord symbol matches its parent
    create a new level; set its value to the new 'generation' of parent symbols

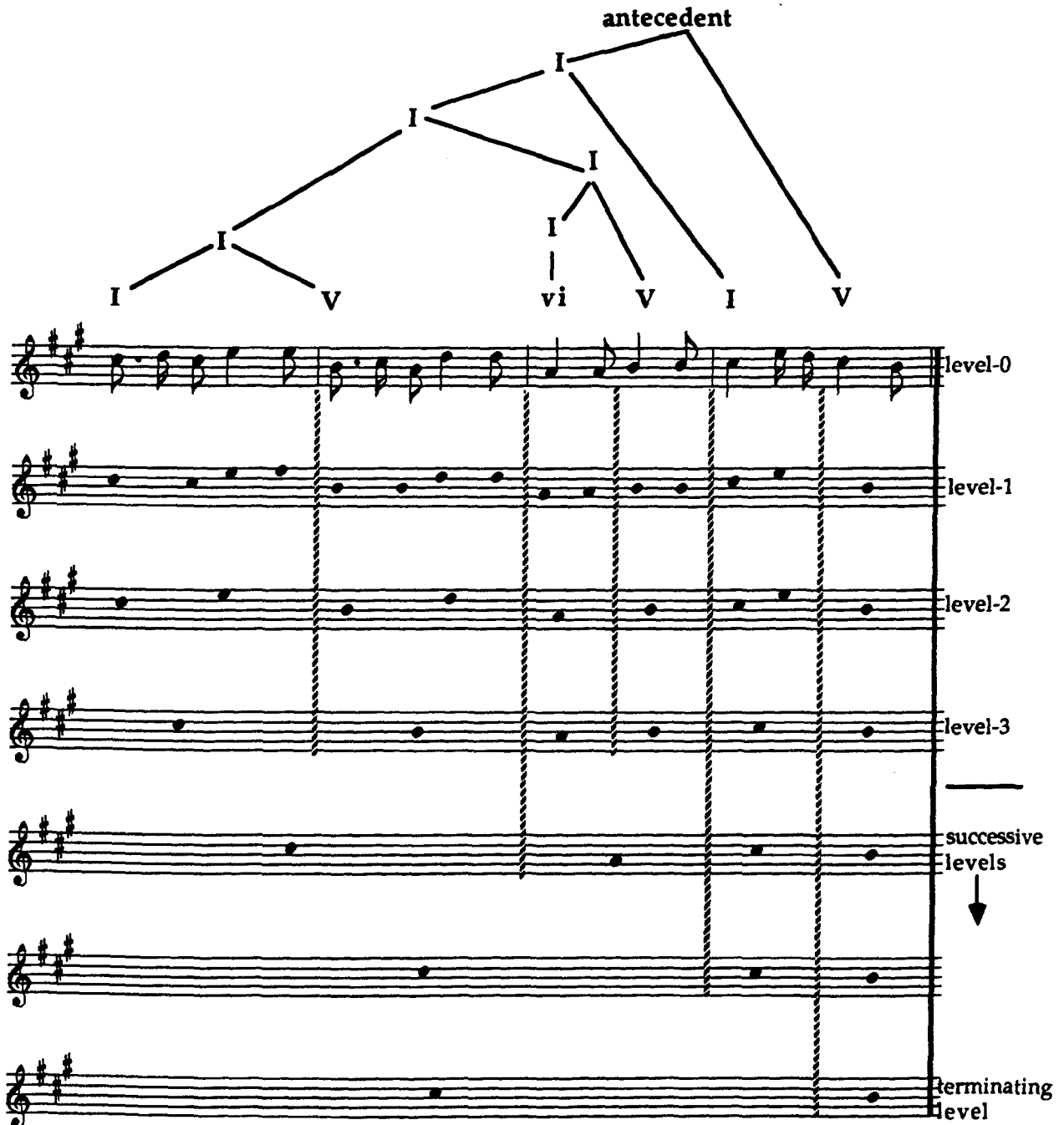
  terminate when the remaining chord symbols match the right-hand-side
    of normal-form
  return
    (level-0 level-1 level-2 level-3 successive-levels)
end.
```

Levels 1 to 3 aim to reduce the melody to a single tone within the scope of each chord function. A simple criterion of 'stability' is used to reduce harmony notes with respect to each other - after non-harmonic and repeated tones have been removed - where less stable tones are those further up in the stack of thirds above a root, and so on. For successive levels, the parse tree not only specifies which tones should be reduced, but more importantly, it specifies which tones are to be compared with each other. Working bottom-up, the 'reduce' function generates a new level from the parent nodes to the previous level, until only the symbols of the normal form remain. Figure 2.5 gives a simple example of how such an algorithm performs a reduction, in which we deliberately choose an example which has been much analysed by other theorists (Schenker 1935/1979, Figure g.72/3; Meyer 1973, p.26 ff; Lerdahl & Jackendoff 1983, pp. 141, 172-173) for purposes of comparison. Note that there are other well-formed parse trees for this set of symbols, which would generate different analyses. It is our hypothesis that the parse tree which displays the greatest 'stability' in terms of deep reductional parallelism will be the musically preferred analysis, and that this indeterminacy between possible analyses can do

justice to cases where grouping perception is genuinely ambiguous.

Figure 2.5

An example reduction of the first phrase of Mozart K331.



The reductional process just described is viewed as the early stages in the description of an objective theory of how reductions can be performed. As such it makes many simplifications, in comparison with the richness of Lerdahl and Jackendoff's theory of time span reductions. These simplifications were made for methodological reasons, assuming that the best way to do justice to the full complexity of musical reductions is to specify initially clear and simple algorithms, which can then be successively refined. To our knowledge, no other methods exist which can form the basis of a computational algorithm for reductions. As Ebcioğlu puts it, "... a formal hierarchical theory of harmony along the lines of Schenker ... is to our present knowledge an as yet unachieved research goal." (Ebcioğlu 1986b, p.192).

At present the model is relatively crude in a number of respects:

- (a) it uses a simplistic criterion of stability of one note with respect to another;
- (b) it needs to use a degree of 'lookahead', for linear parallelisms in selecting the single tone to represent a time-span;
- (c) (related to a and b), the reductions are primarily harmonic, and take little account of melodic and rhythmic factors.

We believe that a *research program* which addressed these and other shortcomings could form the basis of an objective approach to musical reductions. We would not claim that listeners are able to perceive all such levels of reduction in a melody, but simply that our very perception of harmony implies perception of reductions in some sense.

2.4.2.4 Deep reductional parallelism

Once a set of reductional levels has been generated for each of the well-formed parses, pattern-matching techniques are used to search for recurring patterns in harmony and pitch contour at each level. Earlier attempts at using pattern matching techniques (Simon & Sumner, 1968) had failed to pick out musically relevant patterns from the combinatorial debris of patterns which occur in any melody. Our method has the advantages that the parsing process constrains the search for patterns to those which occur between possible well-formed phrase-level groups, and that at reductional levels, 'inessential' notes have been removed to make the patterns which occur much simpler. At present we use a simple set of criteria for preferring one of the well-formed phrase-level boundaries over another: the parse tree is preferred which displays the greatest extent of matching at the highest reductional level, since this corresponds to a match which is in a sense more 'obvious'. If no such match is found for any of the sets of levels, the system repeats the matching process under successive 'limited transformations' until such a decision is reached. A 'limited transformation' is an operation on the pitch-set such as simple transposition or inversion. This forms the weakest part of the model, and an area which requires much further work. Questions which remain to be answered include (a) how can we achieve an ordering of the relative 'distance' of transformations? - is a motif which is inverted thereby perceptibly altered greater than if the motif is transposed? (b) which transformations would apply to any given piece, in order to avoid exhaustive search for the most appropriate?, and (c) how can we avoid a combinatorial explosion in cases where one section relates to another via the application of multiple transformations? The attempt to answer such questions would constitute a research program which attempted to fully formalise Lerdahl and Jackendoff's

notion that for a certain class of tonal pieces, analytically separable components (grouping, metre, time-span and prolongational reductions) coexist in a relationship of *mutual stability* (Lerdahl & Jackendoff 1983, p. 52). Our attempt to use deep reductional parallelism to constrain the space of 'grammatical' phrase level grouping structures therefore rests on the presumption that this elusive notion of stability can be so defined. A number of analyses (programmed, and performed 'by hand', using the theory which we have described) have shown that where a melody *is* of the kind which exhibits parallelism between well-formed phrase groups, then the parse tree which exhibits quite clear deep reductional parallelism is the one which would be preferred by musical intuition.

2.4.2.5 Grouping at the sub-phrase level

This research in modelling grouping at the phrase level suggests that the global factors of 'tonal motion' and parallelism take precedence over grouping preference rules which operate on the musical surface. We would argue that once phrase level boundaries have been determined using the method embodied in AGA, it is feasible to process grouping structures at smaller levels. Although we have not extended our model so far to cope with these sub-phrase grouping structures, we believe that AGA could be extended using a combination of Lerdahl and Jackendoff's well-formedness and grouping preference rules, with sophisticated pattern matching techniques which search for perceptually significant "clues" on the musical surface (Deliege, 1987).

2.4.3 Preliminary assessment: psychology and syntactic processing of music

The overall model of human information processing which is suggested by the system which we have described is one where bottom-up unconscious inference of well-formed harmonic progressions leading to segmentation at the phrase-level, is guided by a *great deal* of top-down knowledge of the specific musical style concerned. In the previous section we discussed the way in which heuristic knowledge is required on a number of levels of processing. At the level of building the 'harmonic lexicon', knowledge is required of the allowable chord vocabulary, harmonic rhythm, and melodic figuration. As lexical items are combined into well-formed progressions, clues on the musical surface give heuristic guidance to constrain a combinatorial explosion of possibilities; and, finally, the search for musically significant parallelisms proceeds at a level of abstraction from the musical surface, guided by possible well-formed grouping structures. At sub-phrase levels, these clues on the musical surface become more important in determining grouping structures.

Tonal music is quite dissimilar to natural languages such as English in terms of local 'lexical ambiguities' - four or five chord functions could possibly be assigned to each metric unit, and furthermore, we don't even know which metric units should be harmonised. This means that in the parsing of even a simple two phrase fragment of a melody, literally *thousands* of substrings are built, which is not at all psychologically plausible. A solution adopted in AGA was to incorporate knowledge into the system in the form of *heuristics*, to limit the building of substrings to those which were likely to lead to musically preferable results, and *genre-specific constraints* on the kinds of melodic figuration and harmonic rhythms which need to be

considered by the harmoniser (these are considered as inputs to the model at present). The latter were based on conventional theory of tonal harmony, and the specification of heuristics was derived from a new theoretical alignment of Lerdahl and Jackendoff's grouping preference rules. There may well be other stronger heuristics which could be specified, but a major problem is to specify heuristics which are 'strong' enough to restrict substrings stored in the chart within human memory constraints, without being so constrained that no suitable parse is found at all. Such heuristics would have to be very strong indeed - in fact, so strong as to virtually constitute the kind of *expectancy-based* knowledge of grouping structures which we discuss in the next section concerning frame-based processing.

A number of limitations of the model have been mentioned in describing individual components, in terms of the amount and direction of processing. There is a further major respect in which the prototype is implausible as a model of human cognition, in terms of the interrelationship between different processing modules, and the direction of processing proposed. In its present form, AGA performs several phases of processing across the entire melody, passing new sets of symbols to subsequent processes. It assigns possible harmonic functions to the entire melody and builds up several parse trees and sets of reductional levels. Clearly, no listener has access to the entire course of a melody in working memory, nor could successive processing components access new sets of symbols in working memory. The unidirectional nature of processing - i.e. the 'generate and test' method - in the AGA prototype is probably untenable as a model of musical grouping cognition. As Jackendoff puts it,

"In musical perception one clearly cannot derive each level of representation in its entirety before going on to process the next level up since that would require listening to the entire piece before even beginning to derive grouping, let alone reductions."
(Jackendoff 1987, §11.9).

To address this problem, the component processes of AGA could be reformulated to produce a model of musical cognition which was more *integrated*, by making the parser the central mechanism of control, invoking dependent harmonisation, reduction and evaluation processes when appropriate. To integrate the parser and harmoniser what is required is to call the harmonisation process when examining a substring's conditions for continuation. To return to our earlier example, supposing the first bar is harmonised by iii or I, matching with grammar rules $I \rightarrow I \quad IV$ and $I \rightarrow I \quad V$ (supposing no rule for iii) would narrow down the harmoniser's task to simply determining whether I or IV was the best candidate for harmonisation of that metric unit. This would make harmonisation *expectancy based*, proceeding in parallel with parsing from left-to-right.

Musical processing may *not* however involve a single unilinear act of processing through time, and may involve some degree of mental rehearsal and *retrospective hearing*. Processing clearly does not wait until the piece is finally over before assigning a structure, but does so through time on the basis of available 'evidence', subsequently performing strictly limited

restructuring of the interpretation if those expectations turn out to be unfulfilled. The top-down element of reductions, and evaluation processes, can therefore operate on the well-formed substrings stored in working memory (the chart) as the parse proceeds, building parse trees and rejecting subtrees which display little reductional parallelism in favour of those which do. Again, this evaluation mechanism is likely to be highly sensitive to the specific properties of the genre concerned, and could easily lead to either over or under constraint, in combination with the set of heuristics derived from grouping preference rules.

The most important limitation of the whole approach concerns the 'closed world hypothesis', which restricts the applicability of the system to a "musical sub-language" (Steedman 1984). AGA only attempts to answer the question

'given that we *know* the melody will be of a particular ('normal') form, how can the system process pitches in order to determine the position of possible boundaries?'

This raises the problem of how the listener *knows* that the piece will be of a certain form in the first place. If the parser were to consider several possible top-level rules simultaneously, then a combinatorial explosion of sub-strings would again arise. As will emerge in the next section, this is a problem which faces any natural language processing strategy, concerning how the perceiver obtains sufficient knowledge from long-term memory and the real world to determine the *context* of an utterance. We offer no solution to this general problem here, but merely describe how it arises in the area of processing musical grouping structures.

2.5 A Frame-based Approach to Grouping Analysis

2.5.1 Introduction

A number of writers (Stoffer 1985; McAdams 1987; Jackendoff 1987; West, Cross & Howell 1987) have suggested that *schema theory* may provide an alternative approach to understanding how a listener's knowledge of stereotypical metrical, harmonic and grouping structures is utilised in musical listening. Their proposals do not, however, include details of how a cognitive model of musical recognition based on schemata could be developed. We describe how schema-based knowledge in the form of *frames* (Minsky 1977) could be used to guide processing as part of a cognitive model of chunking in the auditory domain. Schema theory has been used to explain phenomena as diverse as story understanding (Schank & Abelson 1977), concept representation (Rosch 1975) and as a general theory of memory (Rumelhart & Ortony 1977). Minsky's (1977) notion of *frames* gave new formal rigour to the schema theory¹⁰. Our principal contribution to theoretical research lies in the recognition that the Lerdahl and Jackendoff theory can be reformulated in terms of a more general theory in cognitive science - that of schema theory. Although the AGA component was partially implemented, we did not implement GRAF, but rather described a number of the algorithms for its most important component processes.

¹⁰ See Alber & Hasher (1983) for an influential critique.

2.5.2 Recognition with schemata and frames

Recognition in frame-based systems proceeds by *matching* the contents of working memory at any one time to a set of frames/schemata in long-term memory which are (unconsciously) viewed as relevant to the context of the act of recognition. There are a number of systems which demonstrate such processing mechanisms, two examples being the PIP system (Pauker, Gorry, Kassirer & Schwartz 1984), which used frames to form diagnoses from patient case-studies in a medical subfield, and the GUS system¹¹ (Bobrow, Kaplan, Kay, Norman, Thompson & Winograd 1977), which generated (restricted) natural language dialogues. Its most interesting feature was in the use of two forms of procedural attachment to frames, being 'demons', which are activated automatically when a slot receives a value and 'servants', which are activated on demand from the central control mechanism. We incorporate both these features in the design for the frame-based parser GRAF (GRouping Analyser with Frames). GRAF also uses processing mechanisms which are very similar to PIP, with a much greater degree of procedural attachment. In PIP information about the patient's symptoms are organised in terms of descriptors which may or may not match with the values of 'trigger' slots in the set of frames in long-term memory. "Triggers" are findings which clinicians view as especially relevant, from the mass of data presented to them, to sparking off a particular working hypothesis (a frame representing a disease) which they use to request further data. PIP entertains a small set of such hypotheses at any one time, using probabilistic criteria to select the most likely hypothesis to use to guide selection of new material. In addition, each frame has an associated set of logical decision criteria which specify necessary and sufficient conditions for the truth of the hypothesis, and conditions which imply its falsehood. The GRAF system differs from PIP in that any input information to activate, confirm or disconfirm a hypothesis requires some processing to be done on the input pitches (via procedural attachment), and that several hypotheses drive expectancy-based processing simultaneously.

2.5.3 Design for a frame-based parser: the GRAF system

The GRAF system has been designed to explore the completeness, consistency and psychological plausibility of using the 'integrated processing' version of AGA within a system of frames which represent the kind of stereotypical grouping structures described by Lerdahl and Jackendoff's (1983) 'normal forms'. The first version is designed to recognise the grouping structure of a limited class of *English nursery tunes*¹². These melodies exhibit a small number of possible forms, where the kinds of frame-based knowledge required - such as harmonic vocabulary and melodic figurations - can be easily specified. Furthermore, we have good reason to assume that the kind of melodies which children find easy to remember will be highly stereotypical of many tonal genres. To model phrase structures in these songs, only *four* 'harmonic formats'¹³ are required, corresponding to schema-based knowledge of phrase-level grouping structures. The formats combine two phrase structures, which do not represent the chords which are literally used to harmonise the melody (unconsciously), but which act as 'templates' for the grammar rules described for AGA:

¹¹ See chapter 3 of this thesis for a detailed discussion of this system.

¹² The tunes were largely transcribed from memory, but see Poston 1961, Moorat 1912 and Whyton 1964.

¹³ The terminology is Neisser's: "... a schema is like a format in a computer-programming language." (Neisser 1976, p.55).

$\langle I \ x \ I \rangle$ (= 'C', or 'Circular progression'; x is a variable);

$\langle I \ V \rangle$ (= 'O', or 'Opening progression').

Figure 2.6 lists the harmonic formats required, together with example songs and their chord progressions.

Figure 2.6

Harmonic formats for phrase structures of simple nursery rhyme melodies

(1) CCCC ($\langle I \ x \ I \rangle \langle I \ x \ I \rangle \langle I \ x \ I \rangle \langle I \ x \ I \rangle$);

example:

"The Farmer's in his den"; actual chords: $\langle I \ V \ I \rangle \langle I \ V \ I \rangle \langle I \ V \ I \rangle \langle I \ V \ I \rangle$

(2) O O O C ($\langle I \ V \rangle \langle I \ V \rangle \langle I \ V \rangle \langle I \ x \ I \rangle$);

example:

"Jack and Jill", actual chords: $\langle I \ V \rangle \langle I \ V \rangle \langle I \ V \rangle \langle vi \ ii \ V \ I \rangle$

(3) O C ($\langle I \ V \rangle \langle I \ x \ I \rangle$);

example:

"Looby lou", actual chords: $\langle I \ V \rangle \langle I \ V \ I \rangle$

(4) C C O C ($\langle I \ x \ I \rangle \langle I \ x \ I \rangle \langle I \ V \rangle \langle I \ x \ I \rangle$);

example:

"Baa baa black sheep"; actual chords: $\langle I \ IV \ I \rangle \langle IV \ I \ V \ I \rangle \langle I \ V \rangle \langle I \ ii \ V \ I \rangle$

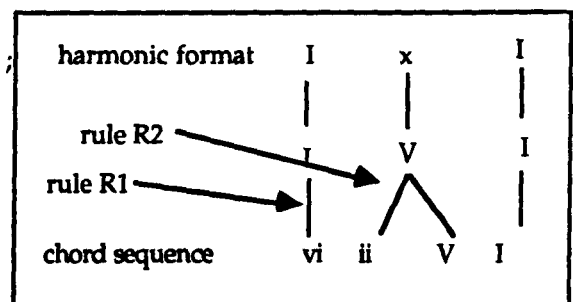
Harmonic formats do not therefore correspond directly to chord progressions in the tunes themselves, but are abstractions from them, via the context-free harmonic grammar described for AGA. They are therefore similar to the 'top-level' rule which acts as input for other rules, used by Steedman (1984). For example figure 2.7 shows how they relate via just two grammar rules for the last phrase of "Jack and Jill".

Figure 2.7

The relationship between grammar rules and harmonic formats

Fourth phrase of "Jack and Jill" = $\langle vi \ ii \ V \ I \rangle$;
harmonic format = $\langle I \ x \ I \rangle$

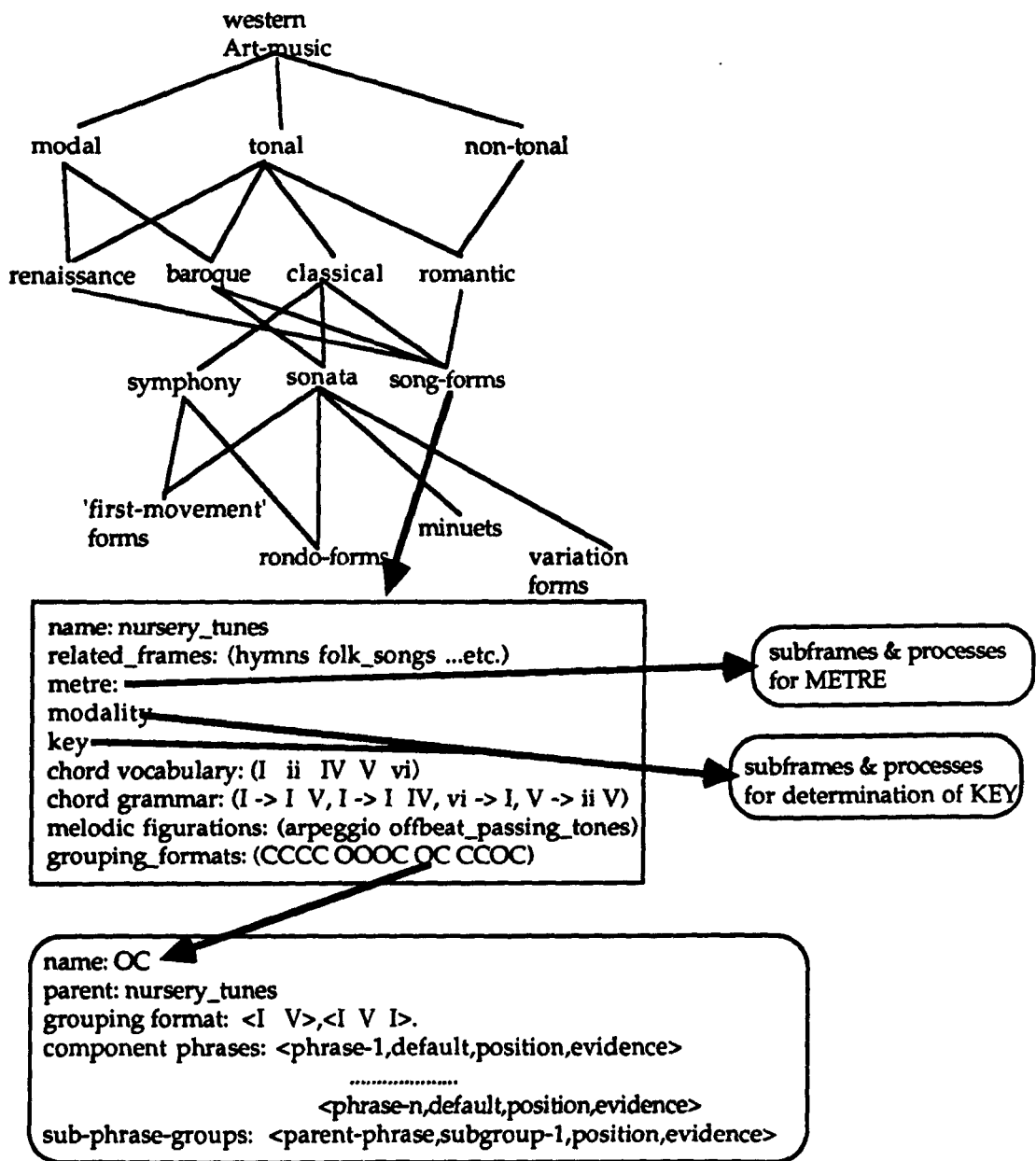
grammar rules required: R1: $vi \rightarrow I$
R2: $V \rightarrow ii \ V$



In order to explain how the knowledge required for the recognition of this relationship is derived, we need to introduce the notion of *context-spaces* (Grosz 1979). For our purposes, a context-space is simply an area of long-term memory which encapsulates all the generalised knowledge specific to a musical genre, which is implemented as the *parent* frame to subframes for possible grouping and metrical structures. Figure 2.8 gives a picture (simplified) of tonal music as a large set of related context spaces, of which our nursery rhyme tunes constitute just one.

Figure 2.8

Diagram of frame system for harmonic phrase formats



Just as in the PIP system described earlier, where each disease frame has pointers to related frames, the nursery rhyme tune frame will have pointers to other closely related genres, such as simple hymn tunes. At any point in the frame recognition process, matching with a particular stereotypical grouping frame can call on the knowledge contained in the parent grouping frame. In GRAF we assume that the key and metre of the piece being recognised is known. For a 'complete' model of music cognition, key and metre would need to be derived from separate

systems of subframes and recognition processes, which could build on existing work in this field (Steedman 1972; Longuet-Higgins 1976; Longuet-Higgins & Steedman 1970; Longuet-Higgins & Lee 1983).

We explain the process of how GRAF could recognise a melody by working top-down through the simplified algorithm shown in figure 2.9.

Figure 2.9

A simplified algorithm for the top-level control structure of the frame-based musical grouping structure recogniser 'GRAF'

```

function:  frame_recogniser
inputs:    *melody* [a list structure]
           *context_space*
           [a system of frames relevant to genre of *melody*]

initialisations:
    *wm* <- nil
    *expectations* <- nil
    *current_position* <- 0

algorithm:

do
    exit when *current_position* is last note of *melody*
    return value of *wm*
    if trigger_clue? at *current_position*
        and matches? (invoke_harmonic_processing
                       *current_position* for each in *context_space*)
        then push matched frame(s) into *wm*

    if *current_position* an expectation in *wm*
        then fill in slot confirmed for each matched frame
            and generate expectations for next slot for each in *wm*
        otherwise, remove disconfirmed frame(s) from *wm*

    generate_expectations for each matched frame in *wm*
    increment *current_position*
    if *wm* limit exceeded, forget some frames

```

The system steps through the melody from left to right, responding to significant 'clues' on the musical surface (potential phrase boundaries) by invoking the revised AGA harmonic processor, and subsequently generating expectations concerning the positions of those boundaries by activating hypotheses (frames) concerning the likely form of the melody. Each frame contains a *harmonic format*, as described in figure 2.6, which represents a possible schematic phrase structure. Matching with the phrase components of these formats is determined by limited harmonic parsing. Several hypotheses are retained in working memory, and are 'filled in' when expectations are confirmed, and rejected when disconfirmed or if working memory limitations

are exceeded. The algorithm reads as above (functions are explained subsequently).

The function 'frame_recogniser' takes a melody represented as a list of notes, and a set of frames (a 'context space') relevant to understanding that melody as inputs. Initially, there is nothing in working memory, no expectations have been generated as to likely phrase-boundary positions, and we start at the first note of the melody. The function executes the following series of steps in order, repeatedly until we reach the end of the melody, when it returns a list of its hypothesised grouping structures (frames with values in the slots), which is in order of preference. It checks whether the current position is a *trigger_clue?* (a position where grouping preference rules strongly suggest a possible phrase boundary). If it is, then it checks each frame in the context space to see if the phrase boundary could be construed as matching (*matches?*) one or more of its slots for phrase boundary positions, by applying the AGA parser to check which of the formats (O or C) the section of music could be rewritten to (*invoke_harmonic_processing*). For any that are matched, it puts the frames into working memory as current hypotheses. If the current position is one for which one or more frames have generated some expectation, then it fills in values for frames whose expectations were confirmed, removes those which were disconfirmed from working memory, and generates expectations for those confirmed frames. For all new frames added this cycle, the function generates their expected phrase boundary positions in working memory. If the limits on the number of hypotheses which can be simultaneously worked on have been exceeded, it *forgets* those which have the least confirmatory evidence. The function then moves on to the next note in the melody.

Triggering processing

The predicate *trigger_clue?* calls the set of Lerdahl and Jackendoff grouping preference rules described for AGA, implemented as functions rather than data structures. Just as a clinician does not attempt to form a hypothesis as to the disease from a single observation, but waits for some kind of key finding or 'trigger' (*re* the PIP system), the idea is that the listener waits for an especially significant 'clue' or indication as to a phrase boundary before matching with phrase-boundary schemata. Depending on the genre concerned, this involves the assumption that possible boundaries which are reinforced by several rule applications (Lerdahl & Jackendoff's grouping preference rule 4, 'Intensification') are better candidates to trigger processing, since processing on every single preference rule indication would invoke far more processing than is psychologically plausible.

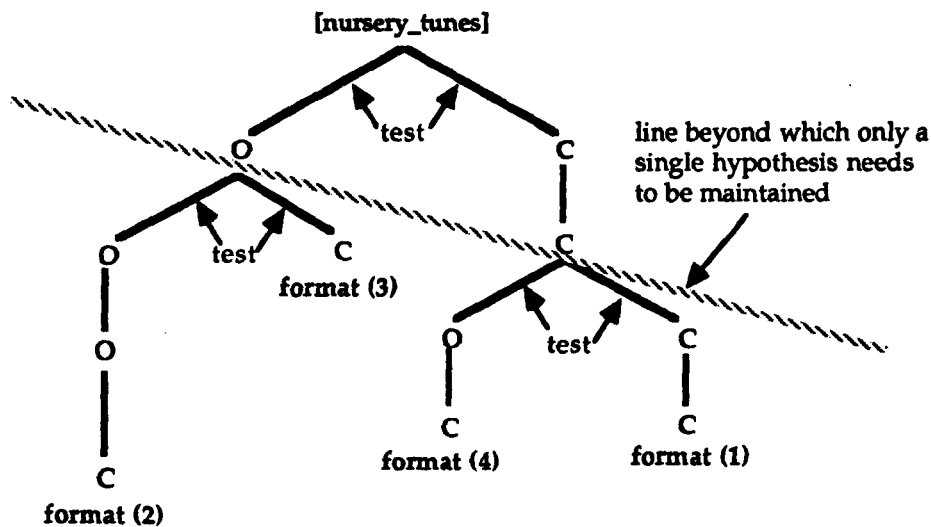
The matching process

The *matches?* predicate checks which of the two possible 'harmonic formats' ('O' and 'C') the possible phrase unit can be construed as, which involves calling the harmonic parser (*invoke_harmonic_processing*) used in AGA with the proposed amendments. This represents a limited departure from the strictly left-to-right processing of the melody in time, and depends on the extent to which working memory can support mental rehearsal and retrospective hearing. Given an elapsed section of the melody on which to work, the function can call on the knowledge residing in the parent frame (see figure 2.8) to constrain the task, so minimising memory and processing demands. In the case of our nursery tunes, we need to consider only two simple melodic

figurations, a very limited chord vocabulary, and just four grammar rules. This should lead to a clear decision as to which of the two phrase formats (O or C) is appropriate (or neither) and hence which of the four larger level combinations (see figure 2.3) will be matched at this point. For example, if the phrase can be rewritten as 'O', and the previous phrase was C, then the format is number 4, CCOC. If a slot gains a value in this way, then rather than the overall control process pushing the frame hypothesis into working memory, a procedure which is attached to that particular slot (called a 'demon') is invoked automatically, so that it sends a particular instance of the frame into working memory.

The matching process is in fact a little more complicated than this, in a way which has important implications for the psychological plausibility of the GRAF system. As described so far the process of matching frames in the context space implies that a frame would take more time to recall (hypothesis formation) the further along the list of frames it is. In fact, this is nothing like the recall process in humans, who, for example, take the same time to recall any word in a list which they "know" (i.e. have overlearned), regardless of its position in the list, assuming that the items are independent in the sense of not being known on the basis of some rule-set which generates one item from another. A better (although still problematic - see Johnson-Laird 1988, pp. 151 ff) model of recall is to use a *discrimination net* (Feigenbaum 1963), which is in fact used in PIP. A discrimination net is a way of structuring data in memory in a tree structure, so that recall time is very fast by the method of "divide and conquer". If each phrase format (consisting of perhaps four subphrases) was represented individually, and stored in a simple unstructured list, then the matching process would have to search this list serially, which would be very inefficient, slow, and psychologically implausible. In a discrimination net, each component phrase of the formats is represented as a node in a decision tree, so that recognition of a complete phrase format would involve simple and more efficient binary search through a network which represented all phrase formats relevant to the current musical context. In GRAF this network would take the form of each phrase format having 'pointers' (the name in a slot) to other frames which could continue the phrase format found so far, so that each of the four phrase formats represented in figure 2.6 would have a unique recognition path in the tree. Figure 5.10 shows this discrimination net, from which it is not difficult to see that if our domain is limited to just four possible phrase structures, with the matching (*matches?*) predicate performing a test at each node, then not only is this process greatly speeded up, but once we are past the point marked with a dotted line in figure 2.10, only a single hypothesis need be maintained.

A discrimination net for the four possible phrase formats in the context of English nursery rhyme melodies



(Discrimination test =
'do harmonic parse (AGA) for current segment of melody')

Generating expectations

Expectations are generated simply by the *default values* in the slots for the remaining unfilled slots in a frame for a phrase-structure. Default values are simply those which are initially assigned unless we receive some information to the contrary. For example, if the first phrase boundary occurs after the first two bars, then we may assume as a 'default' that the second boundary will occur after the next two bars, assuming even-sized groups, unless there is some indication of another phrase boundary in that vicinity, which invokes harmonic processing leading to the disconfirmation of that default value. Again, we believe that the nature of these 'default grouping structures' is encapsulated within an area of Ler Dahl and Jackendoff's theory (grouping preference rule 5) which prefers symmetrical and even-sized groups. Although this is not always the case, we feel that it captures the notion of default grouping

¹⁴ See Charniak, Reisbeck & McDermott (1980), Alpay (1987), for a discussion of the implications of this way of structuring the frame system.

positions, which can be simply calculated from units of the phrase-groups recognised so far by the system at any point. Where this is not the case, the boundary should be detected by the triggering of significant surface boundary clues.

Forgetting hypotheses in working memory

In the restricted domain which we have described, it would be possible to retain *all* the possible hypotheses in working memory. When the system is extended to larger contexts this becomes more and more psychologically unrealistic, and we need to find some kind of criterion for deciding which currently active hypotheses should be lost from working memory. The PIP system used probabilistic measures of the relative likelihood of diseases represented by frames. At present we have no firm answers to this problem, but it is likely that recency effects, and a measure of confirmatory support may play a role. In any case, structuring the context-space in the form of a discrimination net greatly ameliorates these problems in its use of just two phrase formats ('O' and 'C'). In addition to forgetting frame hypotheses themselves, only the values of the most recent slots should be retained in order to do justice to the ongoing process of musical perception.

2.5.4 An example grouping recognition with the GRAF model

The following is a simple *hand-simulated* example of the recognition model proposed in GRAF. The example is the first section of the children's song 'Baa Baa Black Sheep' (shown in figure 2.11), which is necessarily brief, but illustrates the main features of the model.

Assume we have only two frames in the relevant context space for the genre of the melody, which contain the grouping structures

Frame-1: 'C C' ('<I x I> <I x I>'), and

Frame-2: 'C O' ('<I x I> <I V>'),

and the following grammar rules:

$x \rightarrow V$

$x \rightarrow IV$

$IV \rightarrow I$

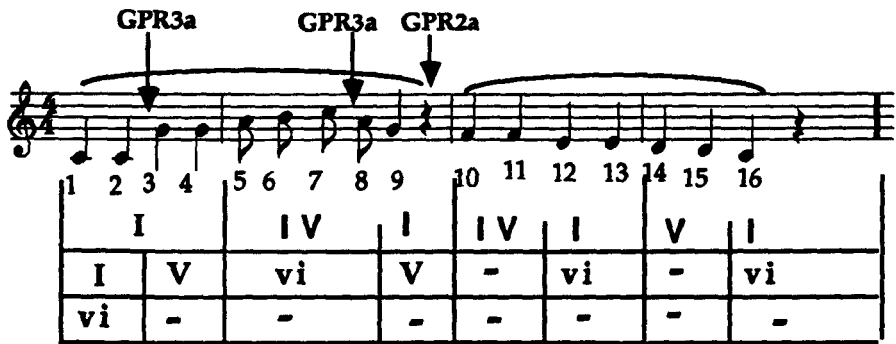
$I \rightarrow vi$

$I \rightarrow I \quad I$

$I \rightarrow IV \quad I$

Figure 2.11
Hand-simulated example of frame-based grouping recognition

Baa Baa Black Sheep



In figure 2.11, possible chords which could harmonise segments of the melody are marked below the melody (these are assigned during parsing, but are marked on the diagram for reference), and points where grouping preference rules suggest possible boundaries are marked above it. GPR2a predicts a boundary given (roughly) a rest between consecutive notes, and GPR3a predicts a boundary after an interval leap which is greater than surrounding intervallic distances. Each note is numbered for reference. The program scans the melody from left to right, checking for 'trigger clues' after each note, and checking whether the position corresponds to expectation values in the next unmatched slot of frames in working memory. When it reaches position 2-3, GPR3a triggers a possible boundary at this point. The program invokes the AGA harmonic parser on the section before this boundary (from note 2 back to the beginning of the melody, since no frames are currently in working memory), and 'lexical analysis' assigns chords vi or I. Since neither of the two possible top-level rules ($C \rightarrow I \ x \ I$; $O \rightarrow I \ V$) can be expanded to these chords, this segment can not function as a phrase unit, and no phrase units of frames are matched. At position 7-8 the interval leap again triggers a possible boundary, but again, examination of the grammar rules available shows that the segment 1-7 cannot be rewritten to a top-level phrase unit. At 9-10 the rest triggers a possible boundary. This time it can be rewritten to a 'C' unit, since the top-level rule $C \rightarrow I \ x \ I$ rewrites to $I \ IV \ I$, using the rule $x \rightarrow IV$, which matches a harmonisation of the segment shown in figure 2.11. The context space is now searched for frames whose next unmatched phrase unit is C. Since no units are currently matched, C matched with the first unit of frame-1 and frame-2, so they are both transferred into working memory, with the value '0, 9-10' inserted as a slot value for the unit. Since a phrase unit has been matched after two metrical units (two bars), the 'default' assumption of symmetrical phrases generates the expectation that the next boundary will be after the next two metrical units, so the value '9-10, 16-17' is input into the appropriate *expectation* slots for the subsequent C and O slots for frame-1 and frame-2, respectively. No further putative boundaries or expectations are triggered until position 16-17, which matches with the expectation slots of C

for frame-1 and O for frame-2. Parsing reveals that C is matched in this case ($C \rightarrow I \times I$ rewrites to a harmonised chord string, via $I \rightarrow IV I$, giving $IV I \times I$, then $x \rightarrow V$ giving $IV I V I$), and so frame-1 ('C C') is matched. Since we have reached the end of the melody, frame-2, expecting O is disconfirmed, and so frame-1 is the recognised phrase structure, marking the phrase boundary at 9-10, and having the final complete harmonisation $\langle I IV I \rangle \langle IV I V I \rangle$.

2.5.5 Summary of GRAF

In GRAF schemata for stereotypical grouping structures in a restricted class of tonal melodies are represented as *frames*, each of which contains a representation of a set of combinations of two basic type of phrase. The two type are 'opening progression (O)', which begins on chord I and ends on chord V ($\langle I V \rangle$) and 'closing progression (C)', which begins and ends on chord I, with either chord V or IV between the two ($\langle I \times I \rangle$). For example, a frame might represent the stereotypical phrase structure 'O C O C'. For English nursery rhyme melodies, just four of these combinations can characterise the whole genre. We claim that *recognition* within a genre can be modelled as a combination of the listener perceiving significant changes on the 'musical surface' (such as a change in dynamics, or a silence), which suggest possible phrase boundaries, and limited harmonic parsing which *discriminates* which of the possible phrase types (i.e. 'O' or 'C') the current segment of the melody can be parsed to (if any). This represents limited retrospective hearing. Significant changes on the musical surface can be modelled as strongly reinforced applications of Lerdahl and Jackendoff's (1983) grouping preference rules, and the 'current segment' of the melody is the section between the present putative boundary and the previous boundary with respect to any grouping schemata. An additional 'trigger' to discrimination by this parsing technique is the representation of default 'expectations' in frame slots - for example, suppose the first two bars of a melody are recognised as an 'O' phrase type, then an expectation is generated that the next phrase group will occur after the next two bars. These expectations relate to the Lerdahl and Jackendoff 'well-formedness' rules for grouping. Each of the phrase types in a frame acts as a 'top-level' rule for the parser, and a melody is recognised by a grouping frame when all of its constituent phrase types are so recognised.

2.5.6 Limitations of knowledge-based approaches

The major limitations of this approach concern

- (i) the process of determining the context of the current listening situation (which frames are relevant?);
- (ii) the matching process;
- (iii) extensibility and generality of frame-based systems; and
- (iv) the issue of 'informal reasoning'.

(i) The algorithms described for the GRAF system considered the problem of how we recognise the grouping structure of a melody, given that we know the context of the listening situation, i.e. a hierarchical set of frames relevant to English nursery rhyme melodies. As Charniak (1982) points out, we do not really know how people combine 'low-level' cues in an environment to activate knowledge stored in long-term memory which is relevant to that context¹⁵. Given

¹⁵ Charniak gives the example of the sentence "As the boy walked down the aisle he took a can of tuna fish from the shelf and put it in his basket", where we have no trouble deciding that the context relates to knowledge of supermarkets.

that there are no known limitations on long-term memory, this is a non-trivial problem.

(ii) The problem of matching arises from the diversity of the objects subsumed under a particular schema, where the detailed properties of one object may be carried over to another for which they are not appropriate. As Winograd and Flores (1986) point out with respect to this problem,

"... if we look at the literature on frame systems, we find a mixture of hand waving and silence."

(Winograd & Flores 1986, p.117).

(iii) It is clear that to the extent to which a system relies on prestored knowledge as to what to expect as input from an environment, its competence can not extend beyond that finite domain. This problem of the highly knowledge intensive nature of many cognitive skills is termed the "real world knowledge" problem in AI research, which is that intelligent human agents are able to draw on the kind of extensive real-world knowledge - from memory and dynamically processed from the environment - which essentially 'closed world' computer-based models are not able to do. We do not at present understand how the vast knowledge of tonal music which most individuals unconsciously acquire in the course of their lifetimes, could be adequately represented in a frame system. Research over the past decade in knowledge-based systems has indicated that to do this is would represent an enormous feat of knowledge-engineering.

(iv) "The frame intuition can be implemented only in a system that does informal reasoning - one that comes to conclusions based on partial evidence.." (Winograd & Flores 1986, p.117). GRAF would need to use informal reasoning of this kind both to specify reasons for preferring one possible hypothesis over another under conditions where working memory is overloaded, and in the assumption that once a hypothesis has been disconfirmed at *one* position in the melody, it will not turn out to be confirmed at some subsequent point. It is possible to amend the design of GRAF to avoid this, by retaining hypotheses which have their expectations disconfirmed in case they turn out to be satisfied later, but at the cost of additional memory. The problem is to achieve the right balance in the trade-off between conserving memory and processing resources, and committing the system to conclusions too early, which may subsequently turn out to be disconfirmed. It seems quite plausible that this accurately represents the state of affairs with human musical cognition, where the grouping of a piece may turn out to be fully understood only on *rehearing*, even though we may gain an initial and sketchy conception of locations of group boundaries on first hearing alone.

Finally, we need to account for perception of grouping at levels below that of the phrase. In terms of our description of GRAF we hypothesise that these levels are most likely to be decided by retrospective hearing and mental rehearsal, once the larger-level phrase unit has been confirmed. It also seems likely that at such levels, pattern-matching under 'limited transformations', such as inversion and transposition, and at successive reductional levels, plays a larger role than exclusively harmonic factors. Given that the area over which pattern matching is constrained in this way, it is easy to see how the algorithm of figure 2.9 could be augmented to include this additional processing as part of the *invoke_harmonic_processing*

function, in a way which did not make inordinate memory and processing demands (as described for the integrated processing version of AGA).

2.5.7 Critical assessment of syntactic and recognition-based approaches

The syntactic processing (AGA) and frame-based recognition (GRAF) approaches to musical grouping perception face precisely the dilemma which has been identified for language understanding models. Greene (1986) succinctly expresses the dilemma as follows:

"... language understanding models are faced with a dilemma: either to specify detailed rules for selecting syntactic categories and word meanings or to rely on world knowledge for resolving ambiguities. In the former case, there are too many sentence interpretations to choose from; in the latter there is the danger of going for only the most probable interpretation. In either case, there is an explosion to deal with, making it difficult to select from many possible word senses and grammatical constructions or from a vast number of knowledge-based inferences."

(Greene 1986, p.144).

In the GRAF system we have argued that the best response to this dilemma is an *integrated processing* approach, which is resource limited, and where the syntactic processing approach is guided by knowledge represented as frames. Given sufficient effort, it would clearly be possible to represent the extensive amount of knowledge required for understanding musical grouping structures either in the form of processing *heuristics* (the syntactic approach) or as *frames*. Both theories have support in the psychological literature. In the former case, unified theories of cognition (such as the SOAR architecture, Laird, Newell & Rosenbloom 1987) are being developed where a theory of *learning* predicts that intelligent behaviour may acquire sufficient knowledge to yield *direct* paths to solutions, thus reducing search. In the latter case, schema theory is well established and pervasive in the literature. One conclusion which both approaches share, and which merits re-emphasis, is that *intelligent musical cognition requires a great deal of knowledge on the part of the perceiver*. In our discussion of GRAF we have discussed a design of a prototype model which aims to increase our understanding of the role of that knowledge in human grouping perception. In the course of doing so we have generated a fresh set of problems, probably the most important of which concerns the role of contextual knowledge in musical cognition. Given the analogy with natural language processing which we have made throughout, this is hardly surprising given the current focus of research in this area on questions concerning how we use knowledge of pragmatics and context in understanding dialogue.

Finally, we may mention a danger which is inherent in pursuing separate lines of enquiry into music cognition. A complete model for music cognition would involve the detailed specification of how individual processes - grouping, metre, harmony - interact, and precisely what kinds of knowledge would be required to manage the balance between cognitive constraints on memory and processing. If we posit separate 'grammars' for each process, then a set of intermediary representations are required between perception on the 'musical surface' and the understanding

of musical form residing in memory. Existing evidence (Johnson-Laird 1983) does not support the psychological reality of these representations. We need to produce a model which does justice to the way in which analytically separable component processes constrain each other, which is at the same time integrated within a single representation. With our present state of knowledge of these processes, such a larger enterprise would probably be premature.

2.6 Summary and conclusions

2.6.1 Research contributions

The research described in this chapter makes a number of contributions to the research field of *music and the cognitive sciences*. We described specific problems and limitations with the model in the previous section, and indicated ways in which future research could extend the model to address these problems. If we look at the parallel field of cognitive models in computational linguistics we find that interest in schematic matching approaches has been surpassed by an emphasis on the pragmatic aspects of language. To understand the nature of our contribution here, therefore, this research must be understood with respect to the 'state of the art' in music and the cognitive sciences, described in section 2.2 of this chapter. It is quite possible that many of the more recent approaches in computational linguistics may find no parallel in music: we cannot expect *a priori* that many of the features of natural languages which have stimulated research into new formal approaches will find correlates in music. At the time that this research was conducted (and reported in recent symposia and refereed journals), there were major problems with Lerdahl and Jackendoff's theory, and a number of writers had proposed that schematic recognition must play a part in music cognition: none of these writers, however, had developed computationally based cognitive models which showed how *processes* operate in such schematic recognition of musical structures, and how such a model could be integrated with Lerdahl and Jackendoff's theory. The major research contribution reported in this chapter therefore resides in the *integration, formalisation and realignment* of existing theories in cognitive science and cognitive musicology. The detailed contributions of this research are as follows:

(1) *Integrating musical processes*

GRAF describes how the processes of pattern perception, musical reductions, surface grouping features, unconscious inference of harmony and schematic recognition in terms of symmetrical grouping structures, *interrelate* in music cognition. No existing research has the beginnings of such a cognitive model. The model reformulates the Lerdahl and Jackendoff well-formedness rules and 'normal forms' as high level schemata, and the grouping preference rules as surface boundary detectors. Further research would be required to incorporate existing work on perception of key and metre.

(2) *Levels of processing*

Lerdahl and Jackendoff's theory does not distinguish successive levels of cognitive and perceptual processing. GRAF represents these levels as low-level surface boundary detection, and matching with high-level grouping schemata via intermediating resource limited

harmonic processing. Although the model shows in general terms how such levels can be interrelated, future versions should represent surface grouping features at the *acoustical* level.

(3) *Formalising musical reductions*

There are no existing formal models for automatically performing musical reductions: existing models use the computer as a tool to assist the analyst (Smoliar 1980), and do not formalise his/her intuitions. An exception is the Schenkerian analysis model of Ebcioglu (1986), but this model makes no claims to being a cognitive theory of how reductions are perceived. AGA gives an indication of how the theory of musical reductions in Lerdahl and Jackendoff's theory may be formalised, in an initially simplified way. Further research in the directions indicated would be required to do full justice to the subtlety of their approach.

We reiterate, that *not all of the model has been implemented* - as described in section 2.4.1, a number of separate components were implemented in order to explore the feasibility of the approach. The major algorithms, data structures and theoretical bases of the model have been described, in a manner so as to facilitate a possible future implementation.

2.6.2 GRAF as a domain model for Intelligent Tutoring

Finally, we should recall that GRAF is intended to provide a basis for a *domain model* for the dialogue model described in subsequent chapters. We should therefore be clear about the nature of the processes modelled by GRAF, and their relationship is with our model of tutorial dialogue which aims to communicate the beliefs derived from these processes. Essentially, GRAF models the situation where a hearer - a teacher or a pupil - listens to a tonal melody and obtains a set of mental representations of the melody and its phrase structure as a result of partial matching with structural schemata stored in long-term memory, being expected structures for stereotypical melodies within a restricted musical genre. The schemata are assumed to be derived from listening experience in a particular cultural context. The specific choice of a modelling approach was designed to produce a symbolic and 'transparent' model for tutorial interaction.

The formation of such mental representations involves further processes and representations when they are explicitly recalled in a *dialogue*. In this case, we assume that speakers are able to mentally rehearse the melody, to intuit *justifications* for beliefs derived from these representations, and that these beliefs are integrated with *semantic memory* of general musical knowledge as *concept instances*. This would require that on *successive hearings* of the melody, the listener recalls the musical features which indicate to them that certain structures are present in the melody, such as their perception of particular chord progressions, intervals, musical contrasts, and so on. In the present version of GRAF and the dialogue model KANT, described in chapter 5, we have not made these necessary linkages (since the implementation of GRAF was not completed), but have specified the knowledge which GRAF would need to provide to KANT, in terms of recording these justifications as part of the parsing/recognition process. This is plausible if we make the assumption of successive hearings of the melody, and

the very strong assumption that the listener is able to remember the melody perfectly in every respect. The output of GRAF which KANT discusses is therefore a set of *uncertain justified beliefs about musical structures* for a given melody. GRAF would need to be extended so that as a slot value for a particular phrase grouping, and boundary, is successfully matched, we also record the surface and harmonic features which led to its matching (the specific surface feature, and chord progression at this boundary). A model which made less strong assumptions would lead to a an even more fragmentary belief set concerning musical structures than a dialogue participant is assumed to possess in KANT. The cognitive model for grouping perception described here has important implications for its role as a domain model. Since frame-based recognition uses *plausible reasoning techniques*, the beliefs about musical structures derived cannot be assumed to be *certain*, and this has important implications for the cognitive model of teaching dialogues used within the intelligent tutoring system. We would claim that such inherent uncertainty represents a psychologically important feature of the domain, which must demand a more psychologically plausible model of the teaching process. A psychologically plausible model for the domain to be communicated is required if we are to model the student's knowledge, and facilitate effective teaching and learning strategies.

Chapter 3: Computational Approaches to Modelling High-level Human-computer Dialogue

"It is clear, however, that 'A believes that p', 'A has the thought p', and 'A says p' are of the form "'p" says p': and this does not involve a correlation of a fact with an object, but rather the correlation of facts by means of the correlation of their objects"

(Ludwig Wittgenstein, Tractatus Logico-Philosophicus, 5.542, trans. Pears & McGuinness 1961).

3.0 Introduction

This chapter is concerned with critically reviewing approaches to developing computational models of human-computer dialogue. Theoretical issues which arise in the generation of high-level dialogue in the context of Intelligent Tutoring Systems are discussed in the next chapter, which describes a specific approach called 'Negotiated Tutoring'. We shall therefore concentrate here on providing the necessary background to the discussion in the two chapters which follow. The problem of developing theoretical models of dialogue has been approached from a number of different perspectives, in terms of the goals and methods of research. Our review is accordingly quite diverse, and is structured as follows:

3.1 Approaches to Interaction in Intelligent Tutoring Systems

3.2 Explanation and Dialogue

3.3 Argument and Dialogue

3.4 Computational Models of Dialogue

3.5 Conclusion

3.1 Approaches to Interaction in Intelligent Tutoring Systems

The collection of papers on 'Intelligent Tutoring Systems' published in 1982 by Sleeman and Brown summarised nearly all the important work in the field at the time. In introducing the collection, the editors singled out three areas of special interest:

- " 1. Implementation of friendly interfaces and conversational systems
- 2. The contribution of student -modelling work to techniques for induction
- 3. Special purpose inference/deduction techniques."

(Sleeman & Brown 1982, p.4)

There was therefore a heavy bias towards student modelling, and teaching of formal or mathematical reasoning techniques, with only one or two systems dealing with interaction as such. In terms of the first goal, the systems which were said to be concerned partially with "Habitable (Friendly) Natural Language Systems" were the semantic grammar language interfaces of SOPHIE (Brown, Burton & DeKleer 1982), also used in ACE (Sleeman & Hendley 1982), and the dialogue rules used in GUIDON (Clancey 1982). The systems were not, therefore,

explicitly concerned with 'dialogue' as it is considered from the point of view of *pragmatics* in linguistic research, in terms of the structure of connected conversation at a level above that of the sentence. The work of Collins and Stevens on the WHY (Collins 1977; Stevens, Collins & Goldin 1982) system was based on analysis of protocols of human teachers explaining physical processes, and was principally concerned with the structure of *topic* shifts in teaching dialogues. Since the publication of this collection, there seems to be a general tendency in ITS towards a greater stress on the importance of the interaction model, and to make connections with relevant work from a more linguistic perspective (such as the the work of Elsom-Cook on IMPART, 1984). We shall therefore concentrate on describing those ITSs which placed a special emphasis on the interaction model, rather than on student modelling or representation of domain models, although any such system must of course adopt some more or less implicit view about the way the interaction is managed. We use the term 'interaction' to refer to the more or less explicit model within a system which controls generation and understanding of communicative acts. In many ITSs the interaction is considered in terms of a set of *teaching strategies*. Since these approaches usually concentrate on the decisions of a *single* agent - the computer -they are termed *discourse models*, as opposed to models of *dialogue*. Our understanding of the term 'dialogue' is essentially that it must aim to integrate *multiagent* decisions:

"... in generative dialogue, any agent must be regarded as both a producer and an understander, i.e. as a complete agent in interaction with some other complete agent(s)..."

(Petrie-Brown 1989, p.196).

Clancey's (1982) GUIDON system had dialogue procedures which were built around an existing medical expert system, where the purpose was to develop strategies for instructing users who were more or less novices in that domain. Expert systems research has, however, its own independent problems in interacting with expert users, which relate to problems of interaction in Intelligent Tutoring Systems. After the initial wave of enthusiasm in expert systems research, a number of fundamental problems are now recognised (see Steels 1987), in terms of users accepting the system's decisions concerning issues of human importance, when they do not understand how these decisions were reached. This problem has led to a large body of research on 'explanation' in expert systems, which is represented in the series of 'Alvey Workshops' on explanation, discussed in this chapter (§3.2). In those workshops we can observe a general trend towards recognising that explanation is something which occurs in *dialogue*, which cannot be simply reduced to sophisticated 'pruning' of rule-based reasoning chains. We shall therefore discuss issues in explanation and dialogue as they relate to explanation in tutoring. It is clear that explanation from a tutoring perspective has different and additional goals to explanation to experts in a domain, since the very domain within which explanations or justifications are generated may not be at all understood by complete novices.

The third of the issues identified in 1982 by Sleeman and Brown concerned "Special purpose inference/deduction techniques". The major example in their collection concerned the

constraint-based reasoning model in SOPHIE-II & III (Brown, Burton & DeKleer 1982), which lead to further research on 'mental models' for human conceptualisation and reasoning. We discussed models of human reasoning in chapter 2 on the plausible reasoning model embodied in the musical parser GRAF, which forms the 'domain model' of KANT (described in chapter 5 of this thesis). Reasoning as a cognitive process is not identical with the process of its explication in *dialogue*, which we might term *argument*. KANT aims to generate critical arguments concerning its beliefs generated by a reasoning model, and so we review contributions from studies of the 'genre' of dialogue which is commonly termed 'argument'. Argument as it actually occurs in dialogue has not been given much attention in Artificial Intelligence research, which has rather concentrated on formal and logical reasoning (including 'default', 'fuzzy' and other logics). The work reviewed therefore derives in part from the linguistics and philosophy literature, as well as from expert systems.

Finally, there is a relatively small body of research which spans the juncture between linguistics and artificial intelligence, which researchers in ITS are beginning to incorporate and extend in their systems. These approaches attempt to model the dialogue process in general, often with the purpose of applying this to generation of human-computer dialogues. The linguistics literature *per se* on conversation analysis, pragmatics and dialogue is of course enormous, and reasonably beyond the scope of this thesis: the work described here concerns itself with *computational* generation of tutorial dialogues from the perspective of intelligent tutoring systems, within the constraint of those aspects which may be presently implemented (or specified) in a computer program.

3.1.1 Socratic dialogues: SCHOLAR, WHY and TRILL

The Socratic approach to concept teaching depends on representing the subconcepts which underlie or relate to the concept being tutored, in a way which is sufficiently fine-grained so that the student will eventually understand the concept in terms of subconcepts which they already understand. Usually, such systems represent concepts in the domain as a semantic network which shows concepts upon which other concepts depend, the first example of which was SCHOLAR - probably the most famous ITS. The system was developed by Carbonell & Collins at BBN (Boston USA) in the early 1970's, to teach a subset of South American geography, using a semantic network (Quillan 1968). The network was used to guide a "mixed initiative dialogue" towards the subconcepts underlying a student's errors, in the sense that both system and student were free to ask *questions* at certain points in the dialogue. There are a number of interesting features in SCHOLAR, including the 'plausible reasoning' technique used in the semantic network. From the point of view of the dialogues actually generated, they consisted essentially of a single interaction type - question and answer - available to both system and student, where topic shifts were guided by prestored importance tags as subtopics were explored. Two main areas for improvement were considered to be the tutorial strategies used, and the modelling of the student's reasoning processes. Clearly, 'question/ answer' is not the only tutorial strategy which human teachers use, and there is something essentially arbitrary about predefined conceptions of the importance of topics, inserted by the system

designer. This latter point often gave a somewhat disconnected flavour to SCHOLAR's dialogues, as the time allotted to discuss a topic in terms of its importance ran out, and the topic was shifted.

These and other issues were followed up in the later 'WHY' system (Stevens, Collins & Goldin 1982). WHY tutored the physical processes involved in rainfall, and thus represents a shift in emphasis from tutoring declarative concepts represented in a semantic network, to tutoring *reasoning*. Extensive analysis of tutorial dialogues led the authors to propose that knowledge should be represented in the form of *scripts* (Schank & Abelson 1977) which express the temporal ordering of factors which cause rainfall. The implications of this choice of domain representation were pervasive within the system, as the authors recognised:

"The types of misconceptions in a student's knowledge that a system can diagnose are heavily dependent on the knowledge representation in the system"

(Stevens, Collins & Goldin 1982, p. 15)

As with SCHOLAR, the system successively *questions* the student about causal factors, until the point where the student response misses out a 'step' in the scriptal explanation. For example, suppose in the explanation of why it rains in Oregon the student says that the air rises over the mountains so it rains. According to the system's script, a step has been missed out for 'cooling' (the air cools as it rises, so rain condenses). The system now searches a subscript below cooling, with the goal of eliciting this missing step, then pops back up to the main level script. The script thus provides a set of *tutorial objectives*. This representation was derived from analyses of human teaching dialogues in which tutors were said to explain substeps and precursors, to attempt to elicit substeps and to correct these 'bugs'.

The authors mentioned a number of problems which the research uncovered, including :

- modelling alternative ways of viewing these processes - such as a functional perspective;
- interference of students' real-world knowledge; and
- modelling interactions between combinations of bugs.

Collins (1977) summarised the 'Socratic method', and a number of its problems as follows:

"The central notion is to force the student to reason for himself, to derive general principles from specific cases, and to apply the general principles that have been learned to new cases."

(Collins 1977, p. 339)

The emphasis is now on teaching *reasoning skills*, rather than on the specific processes operating in some domain, by the application of a set of rules for what *kind* of question to ask the student. These rules in fact form the definition of Socratic tutoring in their terms. The rules principally concerned teaching the necessity and sufficiency of factors involved, their relevance or irrelevance, and consistency; for example:

Rule 12: Probe for a necessary factor

if

(1) a student makes a wrong prediction of the dependent variable because he has not identified one or more necessary factors,

then

(2) tell him he is wrong, and ask him to formulate a hypothesis about another factor that is necessary.

Other rules concern an 'entrapment strategy' to formulate examples which allowed students to realise their own misconceptions, such as

Rule 5: Form a general rule for an insufficient factor

if

(1) the student gives as an explanation one or more factors that are not sufficient,
then

(2) formulate a general rule asserting that the factor given is sufficient and ask the student if the rule is true.

By the time of "A Cognitive Theory of Inquiry Teaching" (Collins & Stevens 1983), these rules had been reformulated as a set of *strategies* for selecting specific cases to discuss, in terms of high level *goals* of

(1) teaching particular rules or theories ;

(2) teaching students how to derive rules or theories ;

with an agenda based control structure for choosing amongst alternative possible strategies.

The strategies summarise general features of the rules above, in terms of :

(1) selecting more important factors and cases before less important; and

(2) moving from concrete to abstract cases.

A more recent example of the use of 'Socratic tutoring' semantic networks to tutor concepts is the TRILL system ("The Rather Intelligent Little Lisper", Cerri, Elsom-Cook & Leoncini 1988) which is designed to teach the concepts which underlie simple LISP functions. The trigger to Socratic tutoring is a mistaken answer to a question posed by the system; for example

[TRILL]: "what is (cdr l) where l is the argument (swing (low sweet) cherry)?"

[student]: "(low sweet)"¹

Since the question related to the 'cdr' function, the system finds the portion of the semantic network relating to cdr, and asks a question based on the node connected to it, being `remove_the_first_s-expression`:

[TRILL]: "remove the first s-expression from the list (pizza (taste good))"

If the student gets this wrong, TRILL asks a question based on the concept of what an s-

¹ The answer should be "((low sweet) cherry)".

expression is, upon which the earlier question depends, and so on.

A critical review of the Socratic teaching approach faces the problem of defining that approach, since as Collins states

"... there is no very specific body of knowledge about the Socratic method ... In fact, until computers provided us with formalisms for expressing "process models," it is unlikely that anyone would have thought of constructing a specific theory about such a thing as the Socratic method."

(Collins 1977, p. 340).

From the empirical work performed by Collins and Stevens, it seems reasonably clear that questions and statements of this kind do occur in certain kinds of human teaching dialogues. The following questions remain, however:

- (i) in its latest form, Socratic tutoring embodies a teacher who controls the interaction completely, and gives the student little opportunity to express their own goals in the interaction;
- (ii) the system's means of representing the domain is not the only possible one, and it seems that considerable parts of tutorial dialogues are concerned with negotiating a common point of view on the domain;
- (iii) a rule-based formalism is not easily amenable to considering the interaction problem in terms of *plans* to achieve the high-level goals, which would be possible given the approaches present direction of the interaction;
- (iv) there is no means of modelling *how* the student revises her/his beliefs in response to the tutoring rules designed to do this².

3.1.2 Semantic grammars: SOPHIE

The various versions of the SOPHIE (SOPHisticated Instructional Environment) generated a long line of research (Brown, Burton & De Kleer 1982), parallel to the Socratic teaching research conducted by Collins and Stevens. The aim was not primarily to generate tutorial dialogues, but to provide a *reactive learning environment* for basic electronics, in which students could try out their ideas, and receive some critical feedback. From the point of view of dialogue, however, the system contained an interface which has had quite considerable influence on ITS interaction models. Recognising that students are distracted from the actual subject matter being taught by having to word their utterances in terms of a formal language, the SOPHIE designers provided a limited natural language understanding capability in terms of a *semantic grammar*, proposed in the dissertation of Burton (1976). The basic idea was that the student's utterances could be parsed in terms of the semantics of the *domain representation*, from the alternative wordings of measurable quantities at the lowest level, to higher level

² In the TRILL-2 system, Cerri, Elsom-Cook & Leoncini (1988) have proposed that in the context of tutoring programming languages, sophisticated tracing packages (such as Brayshaw & Eisenstadt 1989) may be used to help students in acquiring revised models of a domain.

categories in terms of semantic rules such as

<measurement> ::= <quantity> <preposition> <location>

Further sophisticated features of the interface included the ability to resolve anaphora and ellipsis (eg "It has the value 5.7...") by backward searching of the previous dialogue. The technique provides a robust and friendly interface, but has the drawback that the domain must be limited and completely prespecified. This poses a problem if we wish to envisage using such techniques in future tutorial dialogue systems which enable the system to alter its beliefs in response to utterances made by the student.

3.1.3 Coaching, issues and examples: WEST

The 'dialogue' situation involved in WEST (Burton & Brown 1976, 1982) is not a matter of generating a continuous sustained dialogue, where each utterance is immediately followed by another, but rather one of observing the student's behaviour in a computer-based game environment for an opportunity to interrupt, then deciding whether and how to initiate a coaching/teaching utterance. In order to identify "issues" to tutor, the system compares its own optimal solution for each possible move with the move proposed by the student. A move may contain a number of 'issues', which are the skills which the student is expected to acquire. The additional issues contained in the expert's move contain the set of *possible* issues to tutor, and further possible issues are identified by the 'issue evaluator', which scans the student's previous moves to identify issues in which the student is 'weak' - for example, supposing the student has not 'bumped' very often. If there are no additional issues, the system says nothing. From the possible issues which could be tutored at any point, a set of rules are used to decide whether and how to interrupt, including:

- let the student discover things for himself ;
- don't interrupt too often ;
- don't let the student focus on a small subset of skills ;
- don't tutor on two consecutive moves .

The coaching situation in interactive discovery learning environments presents somewhat different problems from those observed in *continuous* interaction: in the former case, the system has to decide whether and how to interrupt as a result of observing behaviour, whereas in the latter it is assumed *a priori* that the system and the student will converse continuously. Nevertheless, in both situations, the system must consider the problems of *what* to say, and *how* to say it. In WEST the answers given to these problems are respectively,

- discuss the topic about which the student knows least, and
- give levels of hints, with general explanations of issues and examples of expert's solution.

This is not so very different from the situation in continuous dialogue where a teacher may decide to introduce new topics in order of those which could be most easily understood by a student. Furthermore, if we adopt a sufficiently general model of 'high-level' dialogue, then the student's 'moves' in an *environment* can be considered as forms of utterance in dialogue (see §3.4 of this chapter). On this view of dialogue, an utterance does not necessarily have to be linguistic - it could be the drawing of a solution graphically, a musical sound, or making some computer mediated action within a learning environment.

The main problem with WEST is that although the issue recognisers and evaluators define a possible space of relevant topics to talk about - the ones additionally required to explain the system's more optimal move - no principled mechanism is provided for choosing between alternative possible topics (WEST can only tutor one topic at a time, and this is decided by a procedurally embedded 'voting' system), and the system has no mechanism for combining successive utterances into coherent topical (or "issue") successions. Topic shifts are simply determined by the student's moves, whereas in interactive dialogues, both participants contribute to the changing course of the topics introduced.

Many of the research problems arising in WEST also arise in 'Intelligent Help Systems', such as the EUROHELP project (the discourse mechanisms of which are described in Winkels, Breuker & Sandberg 1988). As with WEST, generation of "didactic discourse" is opportunistic or 'occasion driven', where information is given to the user by interpretation of user questions or by identification of occasions when the user exhibits non-optimal performance. Given that an 'information need' is identified, discourse is generated by a 'hierarchical planner' in a hierarchy of pedagogical decisions similar to those used by Woolf (1988), described later in this chapter. Whenever a 'local need' (i.e. a specific need) can be recognised as an exemplar of a 'global need', then the system *teaches*, otherwise it just generates *help*. Global needs are represented as 'didactic goals' in a similar manner to the 'Genetic graph' of Goldstein (1982), where possible didactic goals are those which are closely or 'genetically' related to the user's existing modelled knowledge. Once a didactic goal is chosen, it is successively expanded into a hierarchical structure which ends in 'tactics'. As with WEST, a set of 'coaching principles' are specified, including 'be opportunistic', 'be conservative', 'be concise', and 'signal content'.

3.1.4 Discourse Management Networks: MENO-TUTOR

Woolf's model of tutorial discourse (Woolf & McDonald 1983; Woolf 1988) was initially derived from analysis of errors in novice Pascal programmers, and analyses of human tutorial discourse. A general theoretical distinction is made between *guidance discourse*, in which the expert directs the listener to largely new information, and *reconstruction discourse*, in which the expert has the goal of 'reconstructing' the student's faulty or confused knowledge, "possibly by pinpointing misconceptions through careful diagnosis and argumentation." (Woolf & McDonald 1983, p. 230). It is on *reconstruction discourse* that the research focusses, initially using the knowledge base of programming errors as an example domain. Within such discourse, Woolf identifies a number of "speech patterns" (Woolf & McDonald 1983, p. 231) which are commonly used, such as the "grain of truth" pattern, in which a student incorrect solution contains a partial truth, which the tutor acknowledges and then proposes the correct answer. Once the speech patterns and the common transitions between them were identified, Woolf used this information to "reverse engineer" (Woolf 1988, p. 17) "the structures and knowledge necessary for a machine tutor to have a similar model of the student and to make the same transistions" (Woolf 1988, p.17). The fundamental knowledge structures provided consist of a KL-ONE network of topics to discuss (where each topic has an 'importance' factor), an

augmented transition network of pedagogical states, termed the "Discourse Management Network (DMN)", and a set of metarules which pre-empt default state transitions specified in the DMN. The DMN is organised into a hierarchy of decisions, from the most general decision as to whether to introduce, tutor or complete a topic, down to more and more specific decisions, such as whether to propose misconceptions, give examples, acknowledge 'grain of truth' incorrect answers, and so on. Woolf terms this hierarchy of decision states a hierarchy of "planning levels" (Woolf 1988, p. 23), as follows:

pedagogical planning

choice made about 'tutorial style', such as 'Socratic tutoring', which is made before session begins and which is not reconsidered unless the student is performing particularly badly

strategic planning

choice made amongst schematic tutoring scripts, such as "question the student", "choose a new topic"

tactical planning

choice made concerning specific content or topic and speech pattern with which to instantiate the strategic level decision.

Decisions at these levels are separated from the specific low-level linguistic form of an utterance in a specific natural language which would communicate a tactical choice, such as "give an exploratory question".

We may ask where the knowledge comes from which the system uses to actually choose amongst alternative state decisions at any point in the 'planning' hierarchy. These decisions are represented as Lisp functions in 'slots' in each state node, which look at the student model, and characteristics of possible topics in the domain representation (such as their 'importance') and choose which next state to go to. Such procedurally embedded state transition rules are derived from a combination of "reverse engineering" - i.e. experimenting with the tutorial effect of changing the rules - and the representation of rules based on "Clancey's Dialogue Management rules" (described in his 1979 PhD thesis), which we discuss in the next section. An interesting feature of the model is the fact that the pedagogical decisions are not in fact necessarily constrained to those specified in the DMN: metarules can pre-empt these decisions at all levels in the hierarchy, in certain circumstances. A typical example of a metarule occurs when the student makes a series of wrong answers, for example in the metarule 'IMPLICITLY':

"T6-A. IMPLICITLY - a tactical metarule

From:\$explicit-incorrect-acknowledgement

To: Implicit-incorrect-acknowledgement

Description: Moves the tutor to make a brief acknowledgement of an incorrect answer

Activation: The wrong answer threshold has been reached and the student seems confused

Behaviour: Shifts the discourse from a explicit correction of the student's answer to a response that recognizes but does not dwell on the incorrect answer."

(Woolf 1988, p. 25)

One might think that if we want to make a rule that this state change occurs when the student has made a certain number of wrong answers, then this could as well be represented in the DMN

along with the other rules. In defence of this theoretical separation of state and meta rules, Woolf claims that meta rules introduce a new level of *flexibility* into the pedagogical decision making:

"The key point about this control structure is that its paths are not fixed; each default path can be preempted at any time by a 'meta-rule' that moves Meno-tutor onto a new path, which is ostensibly more in keeping with student history or discourse history. The action of the meta-rule corresponds functionally to the high-level transitions observed in human tutoring Formally, the behaviour of Meno-tutor could be represented within the definition of an ATN; however, the need to include arcs for *every* metarule as part of the arc set of every state would miss the point of our design."

(Woolf 1988, p. 24)

In providing a critical assessment of this work we are not primarily concerned as to whether the pedagogical state transitions represented are in fact those observed in human tutoring - this is a question which could only be decided empirically. The key question is whether this specific representation of teaching knowledge, in terms of ATN state transitions with metarules, and a separation of high-level teaching decisions from their means of linguistic expression is internally consistent and adequately captures the evidence upon which it is based. The meta/state rule distinction is based on the claim that meta rules represent 'higher-level' decisions. Consider, however, the default and meta-rules

default: *question about misconceptions* when all current topics are completed and correctly answered, or have been corrected by tutor

meta: *question about misconceptions* when single wrong answer given

It seems that these two rules could be viewed as alternative teaching decisions at the same level, and it is difficult to see how one is more 'general' than the other. Alternatively, we might say that the default path represents the decision made 'after all has gone well, or as expected', and the metarule the decision made when 'things appear to have gone seriously wrong, and the student is really very confused'. In this sense the rule distinction captures an intuition which could alternatively be represented as a distinction between a *plan* which achieves its goal, and one in which the state has not changed according to the plan as observed during *execution monitoring*.

It must be remembered that the tutorial setting for this discourse model is closer to that of the WEST coaching system, in that it is primarily designed for use as a tutor which comments on a student's interaction with an *environment* - such as a Pascal tutor (Woolf & McDonald, 1983), or a model of a paper mill factory (the 'Recovery Boiler Tutor', Woolf 1988). To that extent, the student is given no control over the interaction - other than the ability to interact with the environment - and is largely constrained to be a passive respondent to the tutor's teaching goal interactions, given that the tutor has chosen to interrupt. The interaction³ is extremely 'one-

sided', both in terms of the student's freedom to pursue learning goals' and to influence the system's chosen teaching goals. The addition of meta-rules gives increased flexibility in terms of extensibility of the system and in terms of increasing the range of possible teaching decisions, not in terms of flexibility in those decisions themselves from the student's point of view.

Finally, we make some remarks in connection with the distinction which Woolf makes between decisions at the high-level pedagogic level, and those which generate a linguistic expression of a decision (such as a decision to question misconceptions). Utterances in human dialogues (such as teaching dialogues) are not all on the same 'level' in the sense of simply making some utterance which communicates some factual knowledge. In coherent discourse there are many utterances which linguists have identified as *metadialogue* (discussed in chapters 4 and 5), which refer to the past and possible future utterances in the dialogue *itself* - for example, "you mentioned earlier that ...". Woolf's DMN representation does include some utterances of this kind, such as 'acknowledgement' of an incorrect response, but it could be argued that such metadialogue utterances are *prima facie* candidates for a separate (meta) level of dialogue decision making, like Woolf's metarules.

3.1.5 Case method dialogues: GUIDON

GUIDON was an intelligent tutoring system built around the MYCIN expert system, which reasoned about infectious diseases. The dialogues generated had the purpose of "transferring the expertise of MYCIN-like systems to a student" (Clancey 1982, p. 202), by the presentation of specific "cases" for patients, in terms of their age, sex, laboratory findings, and so on. Dialogues were "mixed-initiative" in the sense that the student had the ability to propose cases⁴ and ask for 'help', and the system was able to criticise student solutions to cases. After the student has input an initial set of keywords, such as "burned meningitis patient", GUIDON *chooses a case to discuss*, and presents the relevant facts to the student. The student can then ask further questions, , whereupon "GUIDON compares the student's questions to those asked by MYCIN and critiques him on this basis" (Clancey 1982, p. 205). For any particular set of facts in a specific case, GUIDON runs the MYCIN expert system to generate a solution trace. GUIDON then compares this solution with the solution input by the student, and uses difference between the two, together with the difference in questions asked, as the basis for generating the interaction. A further goal of the system - in addition to transferring the knowledge base - is to inculcate MYCIN's *reasoning methods* into the student:

"The criterion for having learned MYCIN's problem-solving methods is therefore straightforward: when presented with novel, difficult cases, does the student seek relevant data and draw appropriate conclusions?"

(Clancey 1982, p. 205)

³ In terms of the distinction between discourse and dialogue stated in §3.1, Woolf's system models *discourse* rather than *dialogue*.

⁴ This feature was proposed but never implemented in GUIDON.

Clancey recognised that the MYCIN rules as they stood were not a sufficient representation for communication to a student. He therefore added two further levels of knowledge, which had direct relevance to tutorial rules - a "support level" of textual justifications for individual rules, and an "abstraction level", which organised rules into patterns, based on common factors occurring in their antecedents. A key factor of the design was the separation of domain rules of MYCIN from the *teaching rules* of GUIDON. The key factor in determining how GUIDON conducts the dialogue is the constraints provided by the specific case considered: the MYCIN solution trace provides a series of tutorial "goals" to be discussed, in terms of the rules which lead to that specific goal being concluded. In very simple terms, the topics of the dialogues generated by GUIDON are those found in the MYCIN reasoning trace, and the subset of rules which led to this conclusion, which are organised into an AND/OR tree. This data structure of facts concluded, and the rules which led to their conclusion, fundamentally guides the case-based dialogue. Given this structure, "only one goal at a time is considered" (Clancey 1982 p. 208): *discussing a goal - i.e. fact - means discussing the set of rules which concluded it*. Given this situation, a number of problems arise:

"For many goals there are too many rules to discuss with the student; how is the tutor to decide which to present and which to omit? What techniques can be used to produce coherent plans for guiding the discussion through lines of reasoning used by the program?"

(Clancey 1982, p. 208)

These are the problems that the *tutoring rules* (T-rules) of GUIDON are designed to address. Tutoring rules are invoked in 'packets' by *discourse procedures*. An example of a discourse procedure might be "CASE-DISCUSSION", which tells the system to discuss a particular case and calls a t-rule packet with which to do this.

Our critical review of GUIDON relates to the stated aims of the system, whether it achieves these aims, and the theoretical model of dialogue embodied in the teaching component. Given the stated aims of GUIDON of inculcating the reasoning methods of an expert system, we now know that these are not the methods which humans use (Johnson-Laird 1983), and hence we should not be aiming to teach them to students. Even the experts themselves do not actually use such rules in this way, since they are derived from a *post-hoc* articulation of rules. For example, it is now recognised in the field of medical expert systems that doctors use 'plausible reasoning' methods to form diagnoses - an example of such a system being the PIP system (Pauker, Gorry, Kassirer & Schwartz 1984) described in chapter 2. This led Clancey to propose additional layers ('abstraction' and 'justification'), and Clancey's latest work concentrates on mental models and use of graphical interfaces (Clancey 1988). Clearly, expert system knowledge bases⁵ do not represent a 'deep' enough model of human knowledge (Steels 1987) to serve as a sufficient basis for tutorial communication of knowledge.

⁵ We are not making a completely general criticism of production rule models of cognitive processes, such as the models developed in the work of Anderson (1983).

The 'case-based' method uses specific examples in the domain of medical reasoning to teach rules and reasoning. The final problem which Clancey stated in conclusion (in his 1982 paper) was that of developing automatic generation of a 'case syllabus'. The tutoring problem to be considered should not be simply that of transferring rules exemplified in given cases, but that of organising a rational succession of cases to consider, in terms of a coherent plan to teach domain concepts embodied in successive cases. In discussing a specific case, GUIDON *constrains* the student to follow the system's and/or reasoning tree, and no initiative or freedom is given to the student to influence the dialogue away from this to discuss general issues, concepts and reasoning methods in the domain.

Finally, we need to consider the representation of teaching rules in GUIDON. The specific rules have been quite influential in ITS research - such as in the work of Woolf (1983; 1988) described earlier - and it seems that they embody many correct insights about how to organise the discussion of specific examples. What we lack is a more global conception of tutorial dialogue, which manages the connectivity between cases, and gives greater student freedom and initiative. One of the problems with the representation is its essential 'flatness': teaching rules do almost all the work, and the distinction between discourse rules, teaching rules and dialogue patterns seems somewhat unprincipled⁶. Teaching rules in fact do all of the following (and more):

- select discourse patterns for discussing rules;
- choose new parts of the and/or tree to discuss;
- choose rules to discuss;
- measure the interestingness of rule;
- select packets of further t-rules;
- update the student model and focus record;
- guide discussion of a goal.

One of the purposes of implementing an intelligent tutoring system is to develop a clear theoretical model of the teaching process, embodied in a computer program which itself embodies that theoretical model, rather than to 'engineer' a system which may work, but we do not fully understand how. In GUIDON the representation of teaching knowledge lacks the kind of conceptual distinctions which we might expect, between diagnosis, topic choice, and an abstraction hierarchy of tutorial decisions. The GUIDON research was extremely valuable and influential in ITS research in that the system needed to be built in order to conclusively assess the claim that suitably modified expert systems provide a basis for intelligent tutoring.

3.1.6 Dialogue and Guided-Discovery Tutoring: IMPART

IMPART is an intelligent tutoring system which was designed to teach a subset of Lisp programming by the method of 'Guided-Discovery learning' (Elsom-Cook 1984). The method centres around the student's exploration within a computer-based *environment* (an idea

⁶ In the 1982 description of teaching rules it seems that teaching rules have further teaching rules as their consequents, the consequents of teaching rules are described as 'discourse procedures', which have themselves discourse procedures within this consequent, and so on.

deriving from work on LOGO, Papert 1980), with the recognition that user's need some degree of *guidance* to supplement the 'free exploration' advocated by the LOGO approach. Given the centrality of this concept of 'guidance' and its *directiveness* or *degree of constraint* in computer-based environments, intelligent tutoring interactions can be viewed as points on a continuum from complete *constraint* - such as in the case of GUIDON (Clancey 1982) - to no constraint - as in the case of LOGO. IMPART - and its version of Guided-Discovery Tutoring - advocates a flexible position between these two extremes, where the amount of guidance given relates to the proficiency of the pupil, and the goal of the interaction is to 'successively withdraw guidance' to the point where the student no longer requires the system's help:

"With a novice, the system performs a large amount of "hand-holding", but as the proficiency of the pupil increases in various areas the tutor gradually withdraws until the pupil is operating in an unadorned programming environment."

(Elsom-Cook 1985, p. 383).

Discourse mechanisms in IMPART are therefore designed to control interactions of this kind, in environments where the tutorial context is quite analogous to that of the WEST system (Brown & Burton 1982). The general interaction problem considered is

"... to take a set of hypotheses about items to teach (from the learning model), some top-down domain specific goals (the curriculum), some rules about educational style some rules about "good interaction", and to blend these into an integrated discourse between pupil and teacher which satisfies the needs of each."

(Elsom-Cook 1984, p. 165)

The discourse system contains four major constituents:

- (1) a conversational context;
- (2) a set of topic controllers;
- (3) dialogue games; and
- (4) a conversational controller.

(1) *conversational context*

The conversational context includes :

- noting what has been said and associated a focus rating with each topic controller;
- the pupil model, hypotheses concerning which help to bring certain topics into focus;
- the set of items referred to in pupil initiated steps.

(2) *topic controllers*

All these factors influence the extent to which specific 'topic controllers' are "in focus" (Elsom-Cook 1984, p. 179). The topic controllers are the fundamental "topic-based" units of domain knowledge in IMPART. They are not, however, simply declarative representations: they have associated procedures which are invoked when the topic controller is given control of the

interaction, which includes activation of "dialogue games", which control the way in which the topic is discussed, and the activation of other topic controllers which talk about issues upon which the first controller depends. This invocation of dependent controllers proceeds until the controller has decisively succeeded or failed in its attempt to achieve its goal, whereupon control is transferred to the original controller.

(3) *Dialogue games*

The "Dialogue games" used are a subset of those used by Power (1979) - discussed later in this chapter - and constitute functional or goal based units of dialogue. An example is the 'info-seek' game - "Attempt to get information from the other participant in the dialogue." (Elsom-Cook 1984, p. 179). An important additional game is "illustrate" - "give more concrete discussion of some topic". Criticising Reichman's (1985) distinction between 'issue' and 'non-issue' dialogue structures, Elsom-Cook argues that 'concreteness' in the interaction is an important feature which should admit of a number of levels.

Each dialogue game has at least one procedure associated with it for achieving its goal (such as IMPARTing some topic). For example, when the "illustrate" game is invoked, with the 'evaluation' topic, it can invoke a dynamic tracing package, rather than describing the topic in English. This is an important point of view on the nature of high-level dialogue - especially in computer-based environments - where a particular decision at the dialogue level can achieve its goal in a number of different media.

(4) *conversational controller*

Given a representation of topic controllers and their respective 'bids' for control of the interaction, the fundamental dialogue generation algorithm is:

- get all bids (topic controller)
- choose a subset which may be given control
- filter the topics and order them, imposing constraints in the form of a set of *tutoring rules*

Iterative repetition of these steps at each point where the system decides to say something, generates the dialogue. Bids are generated by a complex mechanism which takes into account 'intrinsic importance', 'extrinsic importance' (recency and relevance to set of current hypotheses which the student could hold derived from the Bounded User Modelling component), 'level talked and tested' and subconcept structure.

The teaching rules used act to combine possible topic controllers into a rational teaching sequence, and to constrain possible choices of dialogue games for each controller. Examples - which bear similarities to the coaching rules developed for WEST - are

"1. don't discuss a topic for which none of the subtopics have been introduced...4. Do not permit more than 2 occurrences of the same functional goal in succession."

(Elsom-Cook 1985, p.387)

In this way, selected topics and their associated dialogue games are combined into a 'plan' for an isolated segment of discourse. For example,

(function illustrate impart

(argument illustrate action-probe)

(evaluation impart

(value illustrate))

(Elsom-Cook 1985, p. 391)

With respect to problems faced by the system, Elsom-Cook makes the following remarks:

"Note that (theoretically at least, since the full discourse system has not been completely implemented) either participant in the conversation can assume either role in the game. In fact, the internal structure of the discourse games in IMPART is somewhat arbitrary. For example, it is not clear that the above structure [an example of a dialogue game] handles all forms of requirement for action, and many parts of the code for the game are specifically tied to the representation used in the rest of the Tutoring system. The development of more rigorous framework for describing the particular details of these games should be a focus of further research."

(Elsom-Cook 1985, p. 386).

Other features left for further research were the support of "mixed-initiative" dialogues, where the tutor was able to recognise when the user wanted to initiate a dialogue game, and the development of a more systematic mechanism for generating plans.

There are a number of ways in which IMPART 'broke new ground', and was the precursor of a number of present trends in modelling tutorial dialogue generation:

- (1) IMPART was one of the first ITSs to begin to apply work on linguistically inspired models of dialogue, such as that of Reichman (1985), Power (1979) and Levin & Moore (1977).
- (2) Work in computational models of dialogue had included work which concentrated on topic-based structures (such as Grosz 1977, 1979; Reichman 1985) and on goal-based structures (Levin & Moore 1977; Power 1979). IMPART was based on the idea that any satisfactory discourse theory should aim to incorporate *both* of these kinds of structures, which were incorporated as topic controllers and attached dialogue games.
- (3) The guided-discovery approach stressed the importance of the dimension of *constraint* on the interaction, imposed by the ITS, and its role in dialogue.
- (4) The importance of dialogue per se was emphasised in IMPART more than in other systems at that time, which had previously concentrated on student modelling, domain expertise and

'bug finding'.

3.1.7 Summary of approaches to interaction in ITS

The central theme which emerges from this review of some ITS approaches to interaction is the necessity for an ITS to decide *when* to say something, *what* to say and *how* to say it. Given that these decisions can be made on a local level - i.e. for each specific tutorial intervention - there is the further problem of coordinating these local decisions into something which we would term *coherent dialogue*. An important dimension in coherent dialogue is the degree of freedom allowed to the student in influencing the overall course of the interaction. Let us summarise the specific solutions to these questions given by each system reviewed:

SCHOLAR

In the early version of SCHOLAR, the question of what to teach at a given point is answered by choosing the next concept connected to the present in the semantic network in order of predefined 'importance'. The question of how to teach was not reconsidered at each interaction step, but was predetermined by a simple conception of Socratic tutoring as 'question/ answer'. The dialogue is continuous and 'mixed initiative' in the sense that the student is given the possibility of asking questions as well as the system.

WHY

By the time of the WHY system, Socratic teaching had been formulated into a set of rules which in fact *combined* answers to what and how to teach, by giving rules for choosing examples, subconcepts and so on, paired with an interaction style which stated the examples, or asked questions about them. The student was no longer given the opportunity to influence the course of the interaction.

SOPHIE

The interest in SOPHIE from a dialogue point of view was its use of semantic grammars to provide robust a natural language interface for the student, without requiring a full natural language understanding capability. This is important from the point of view of allowing the student a certain degree of freedom in the interaction but means that the system knowledge must be restricted. The dialogue is "case based", since SOPHIE proposes initial problems in a circuit and critiques the student's answers.

WEST

In WEST the emphasis is on identifying *what* to say by differential modelling of "issues" and "examples" to discuss, and on *when* to interrupt the student with a teaching interaction, which led to the development of a set of 'coaching rules'. Fundamentally, the issue chosen is that in which the student is weakest. The issue of how to discuss a topic raised the important issue of level of concreteness of an issue or example discussed, and the giving of levels of hints. The student is given no freedom of interaction.

MENO

MENO focussed on the problem of how to talk about a particular topic, which is initially constrained by the student's interaction in an environment - for example, as in WEST, possible

topics are the 'errors' identified in a student's Pascal program. The decisions to be made about how to interact are made at successive levels of 'planning' in terms of a discourse management network (an ATN), with meta-rules for preempting default state transitions from one pedagogical state to another. The student is given no freedom to interact.

GUIDON

In GUIDON the decision as to what to teach is made at two levels. Firstly, the general succession of topics is constrained to those arising in a traversal of the system's expert system solution trace, and differential modelling with the student solution. Secondly, a set of teaching rules are provided for deciding which rules to discuss and in what order. Given this decision level, the interaction styles for discussing a rule or solution trace conclusion consist of *elicitation* of solution steps by questioning, and simple generation of justification text. The student is given the freedom to ask for help, and so on.

IMPART

In IMPART the decision as to what to say is determined by initial identification of possible "issues" in the student's interaction with a Lisp environment, which are then given priority ordering according to a 'bidding' system, then filtered and ordered into a rational interaction sequence by a set of 'planning' rules. Interaction styles for each issue consist of a variety of possible 'dialogue games'. In the present implementation student initiated dialogue interactions were not supported.

We can observe an initial distinction between those systems which use continuous dialogue to teach (such as Socratic teaching systems) and those where dialogue is initiated by the student's interaction with an environment. In the former the problem of what to teach is constrained by the previous dialogue and the student's knowledge, and in the latter, by the topics introduced by the student's interaction with the environment. Systems usually contain a very limited set of actual interaction styles (with the exception of IMPART), and very few of the interaction methods and decisions are usually available to the student. In a sense, if we adopt a sufficiently general conception of dialogue, the distinction between dialogue initiated in environments and continuous dialogue becomes less distinct: the system and the student may both choose to perform dialogue actions linguistically, graphically or by interacting with an environment.

We have summarised the major research in ITS which places an emphasis on the interaction model. Two other important systems from the interaction point of view are 'DOMINIE' (Elsom-Cook & Spensley 1988) and 'DECIDER' (Farrell 1988). DOMINIE is one of the very few implemented ITSs which provides some solutions to the problem of integrating multiple teaching styles, and of choosing between them at a given point in the interaction⁷, and will be discussed further in chapters 4 and 5. DECIDER proposes a tutorial style called 'self-education', as a method for modelling students' "assumptions" in reasoning according to 'scripts' (Schank & Abelson 1977), and providing feedback (counter-example cases which violate the assumption(s)) which leads students to question those assumptions. The overall style proposed

⁷ See chapter 5 for our proposed solution to these problems.

has implications for the general role of an interaction model in ITSs, but the emphasis is on *student modelling* as opposed to modelling the structure of an *interaction*. For that reason we discuss DECIDER in the final chapter, in terms of a critique of the general ITS approaches of 'self-education' and 'Negotiated Tutoring'.

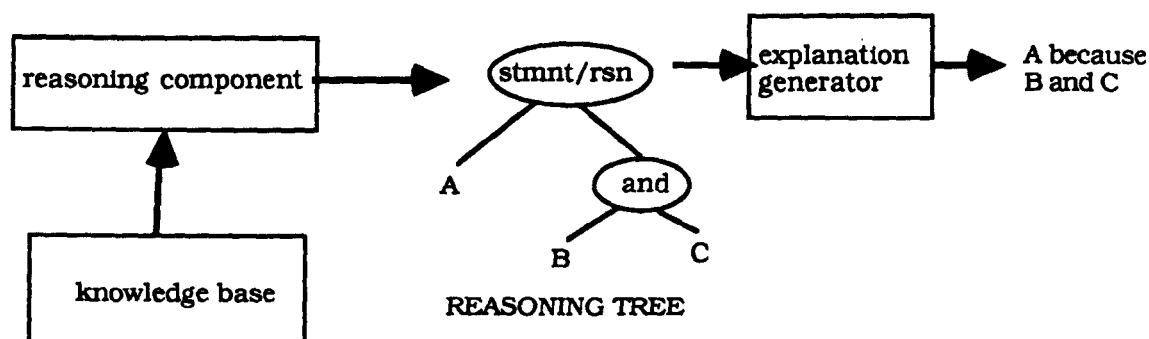
3.2 Explanation and Dialogue

The interest of the Artificial Intelligence community in explanation was initially inspired by the problem in expert systems research that users were unwilling to accept conclusions which were based on reasoning which they did not understand. The problem was then seen as that of generating effective *explanations* of rule-based reasoning processes, which were tailored to the user. Since the result of a question posed to an expert system was an 'answer' output in the form of a reasoning chain, with the rules which led to its conclusion, an initial view of the problem was to see it in terms of somehow 'pruning' that tree to eliminate unimportant or redundant information. We have observed many of the same concerns with pruning reasoning trees according to a set of rules in a review of the ITS GUIDON, built on the MYCIN expert system. In the evolution of explanation research we can see a gradual progression towards the recognition of the necessity to include user-modelling techniques, and a move away from the view of an explanation as a linguistic (or graphical) 'object', which can be somehow effectively shaped and 'given' to the user. The recognition that as part of an explanation a user may wish to pose further questions, and the influence of a similar trend towards applying linguistically inspired work on dialogue, led to the present position that explanation is something which *emerges* from a sustained *dialogue*, which includes negotiation of a common level of understanding between participants. In this section we shall briefly summarise some of the major research which illustrates these recent research trends.

3.2.1 Explanation in rule-based systems

An influential approach to the problem of explaining solution traces was incorporated in the 'BLAH' system (Weiner 1980). BLAH is a deductive problem solver in the domain of US Income Tax law, which interfaces to a truth maintenance system (Doyle 1979), for determining the set of beliefs currently held by the system. Attached to each assertion is a set of justifications derived from the course of a deductive inference, and the aim of BLAH is to present these to the user of the system in the form of a structured and comprehensible explanation. Justification-types correspond to logical connectives of the inference, and take the form of a grammar. A diagram of the system is shown in figure 3.1:

Figure 3.1
The BLAH system (Weiner 1980)



The reasoning tree shown corresponds to the method for explaining the statement (A because (B&C)).

There are fundamentally two ways in which the reasoning tree is adapted to the user:

- (1) it is 'pruned' according to a simple user-model, on the principle that it is not necessary to tell the user something which is already known;
- (2) the pruned tree is presented to the user in a coherent form by the use of an *explanation grammar*.

The grammar is based on analysis and formalisation of human explanations. For example, in explaining choice between alternatives, the differences should be explained first:

"An explanation of why one alternative was chosen over another would be incoherent if the similarities were mentioned first, since this would contradict the user's expectations."

(Weiner 1980, p.71)

BLAH thus contains effective mechanisms for structuring single *textual* explanations in a coherent form, which try to adapt to the user. There is no mechanism for flexibly altering, expanding, or amplifying the explanation in response to user utterances, given that it is always possible that the system's decisions as to what the user wants may be insufficient or mistaken from the user's point of view. It seems that the pruning and form of an explanation could be better adapted in the course of a more extended interactive dialogue.

In the field of medical expert systems, problems of acceptance and explanation of an expert system's conclusions by human experts led to an approach which reversed this situation: instead of the system providing and explaining a solution, the system was to be made more 'user-centred' by initially asking the *human user* for his/her solution, which the system 'critiqued' by comparing it with its own. Critiquing systems attempt to give an explanation of

their reasoning by giving an explanation of the significant differences between their proposed approach and the approach stated independently by the doctor. Two examples of such systems are the ONCOCIN system (Langlotz & Shortcliffe 1983), which is an expert system for applying rules for cancer treatment to patient-data, and the ATTENDING system (Miller 1984), which gives advice on possible alternative approaches to anaesthetic treatment⁸. In both systems, the doctor inputs a suggested treatment plan via menu-driven interfaces, and the system presents a critique of clinically significant differences between its own suggestions and those of the doctor. In ONCOCIN, the system applies production rules to patient-data, and then works top-down through the tree of alternatives of the doctor's plan and that of the system. It stops at the first difference and presents it if it is viewed as significant. ATTENDING uses prestored plans for alternative courses of treatment in the form of Augmented Transition Networks, where nodes represent treatment decisions ('give drug x...'), and arcs are labelled with estimated risks associated with those treatments. Critiquing involves pointing out alternative approaches and their relative risks.

From the point of view of intelligent tutoring, critiquing systems as they stand do not help *novices* to construct plans in the first place, nor do they aim to teach the domain concepts to be embodied in such plans. It seems clear that in order to do this, a more extended *tutorial dialogue* would have to take place, in terms of which we can view a critique as a form of dialogue which is constrained to a single set of utterances from the system and the user. In a tutorial context, a single response to a student's solution would rarely be sufficient: the student would usually want to respond to the tutor's critical response, perhaps to question a factual assumption, request clarification on a conceptual matter, and eventually to submit a revised solution on the basis of negotiation. Critiquing approaches give *expert* advice, to users who are already reasonably expert in that subject domain, but do not give users interactive influence over the form and content of the advice given.

In summary, expert system based approaches to explanation have concentrated on two main approaches:

- (1) pruning the output reasoning tree according to simple user models, and arranging the remaining rules and conclusions into a coherent piece of text according to explanation grammars;
- (2) critiquing the differences between the system's reasoning solution and the user input solution.

The problem with the first approach is that given the way in which human memory and understanding changes as new information is understood, it is unlikely that an essentially static piece of text - however well structured - could be sufficient for explaining a complex piece of reasoning. It is likely that the user would require several interactive exchanges to clarify certain points, and that this new information would cause further changes in human understanding which in turn required further exchanges. With respect to the second approach,

⁸ A similar approach to critiquing a user's explanation for data relating to interpretation of magnetic spectra was adopted in the the ACE ("Analyses Complex Explanations") system (Sleeman & Hendley 1982).

although it is more 'user-centred', each critique is unconnected to further critiques, and the system could have no effective means of understanding the user's existing knowledge and interests within a single exchange. These and other problems have led to an increasing awareness in expert systems research that sophisticated dialogue mechanisms need to be developed, together with an enriched domain representation to support them.

3.2.2 Explanation and interactive dialogues

The 'Explanation and Dialogue' sections of the past four Alvey Workshops (from 1986 to 1988) on Explanation (see references by Draper and by O'Malley) have moved increasingly towards recognising the fact that *explanation requires an extended dialogue* (O'Malley 1987b) rather than generation of isolated utterances. In his review of these workshops in 1988 Gilbert (1988) remarks that:

"Initially, an explanation was regarded as a chunk of information which could be produced by a system to show how it had arrived at a conclusion. By the fourth Workshop, explanation was no longer a 'thing', but a process, and no longer produced by a system, but generated collaboratively in dialogue."

The principal reasons why a more fully interactive dialogue is required concern:

- (i) establishing the knowledge assumed by participants in the explanatory dialogue,
- (ii) establishing 'relevance constraints' (what will be accepted as an explanation by all participants?), and
- (iii) the necessity for establishing mutual goals in the interaction.

Generating such fully interactive dialogues has turned out to be a very hard problem, which needs to draw on work from cognitive science, artificial intelligence and linguistics. The work of Cawsey (1988; 1989) can be viewed as transitional between explanation as generated by grammars, and fully interactive explanation in dialogue⁹. In her work, the problem of turning explanation grammars into comprehensible natural language is approached by viewing each rule as a *planning operator*. For example, the following are some of her rules for explaining an electronics circuit:

Explain -> describe, explain behaviour, summarise

Explain -> type of circuit, summarise

Describe -> type of circuit, function, describe components

Describe -> analogy

The applicability of the left-hand side of a rule is determined by *preconditions* which relate to the domain and to a user model. The relevant operator for this left-hand-side represents the right-hand side as *subgoals*, and has text templates attached for describing the goal. Subgoals are selected if their preconditions can be satisfied, and can themselves be strategies, concepts to

⁹ In terms of our definitions of dialogue and discourse, Cawsey's model is in fact one for *discourse*.

be taught, or other Lisp routines (to call graphics, for example). We can see that given a top-level rule as an initial goal, it can then be expanded using a hierarchical planner (Sacerdoti 1977) within the constraints of goal preconditions, to give a complete 'explanation' to be generated. The final explanation tree is represented as a *goal stack*. As the system executes the goal stack, it pauses after each, to allow the user to interrupt and ask questions. The goal of answering these questions is then pushed onto the goal stack. Goals are executed using text templates, which may include asking the user questions if the system is unsure as to whether a concept is understood according to the user model. After an interruption has been dealt with, the system resumes where it left off in the goal stack, reminding the user of what was being explained before the interruption.

We have described Cawsey's work as transitional between 'explanation as text' and 'explanation as dialogue' because a number of features of interactive dialogue have not been included in the present implementation of her work. Although the student is given the opportunity to interrupt, interruptions are confined to those which are relevant to the system's plan (or 'potential plan', since hierarchical plan nodes are expanded as the explanation generation progresses), and specifically, to a repetition of the previous section of the system's explanation:

"... questions are implemented by producing a menu of descriptions of all the supers of the current goal (the goal last resulting in text or questions) ..."

(Cawsey 1988, p.96),

i.e. the *system's* previous goal. The user is thus constrained to ultimately receiving an explanation in the system's terms, which may be further clarified by interrupted questions. In fully interactive dialogues, each participant may have the possibility of attempting to pursue their own independent goals - providing these are implicitly cooperatively agreed - without the necessity of ultimately returning to pursue the original goals of the other. Dialogue focus occasionally shifts sufficiently in interactive dialogue so that the interrupted participant cannot relevantly resume where they left off, and must plan a new sequence of utterances constrained by the new dialogue state (if they do in fact form such plans).

The research on explanation which comes closest to full interactivity is that of Frohlich (1988), on the application of 'conversational dynamics' to generating explanation dialogues for an expert system concerned with welfare benefits rights. Explanation is viewed as an 'emergent property' of dialogue, which will

"...take care of itself in a system of this kind, if the ensuing dialogue between user and system is controlled in a sufficiently flexible manner."

(Frohlich 1988, p. 114)

Flexibility is provided by allowing both the system and the user to pose and answer *questions* ,

within a system of *turn taking*, based on the work of Schegloff and Sacks (1973). The dialogue is accordingly organised into a series of *turn constructional units* where speakers may each have the opportunity to either self-select or allow the speaker to shift at a *transition relevance place*:

"... the system gains the opportunity to self-select after the user selects a punctuation mark from the dynamic menus used to compose statements. The user gains the opportunity to self-select after each of the system's statements, which are displayed word-by-word at a slow reading speed."

(Frohlich 1988, p. 115)

Given the free provision of turn-taking and a variety of question and answer types, explanation is viewed as emerging from the succession of *questions and answers*, where

"... it is possible to see in principle how users might perceive the answers to follow-up questions as explanations of previous utterances."

(Frohlich 1988, p. 121)

A number of features of this approach deserve emphasis:

- users are given a symmetrical degree of freedom to interact, and a similar range of interaction types to those possessed by the system ;
- there is no user-modelling and tailoring of explanation text, since it is assumed that users can simply ask for what they want to know rather than having the system attempt to infer this:

"If users can ask explicitly for the kind of information they require at any moment in the interaction, the system may not need to work out their requirements at second hand."

(Frohlich 1988, p. 121)

- the work is based on linguistically inspired research in conversation dynamics.

The virtues of the research clearly seem to lie in its emphasis on 'naturalistic' features of human-human dialogues, such as the freedom of each participant to take turns in the dialogue. It is clear, however, that providing such a dialogue structure and giving the user the freedom to ask for information does not solve all problems in human-computer interaction. Research in development of online help systems (such as O'Malley et al, 1984) has shown that users of complex systems are not always able to frame their questions appropriately - in other words, they do not always know what is relevant to their problem, and they do not always know what their problem is. In order for a system to elicit user's problems, sophisticated diagnostic user-modelling facilities, and additional dialogue structures to support them, may well be required after all, together with architectural system components which enable the system to be more than a passive respondent in planning coherent sets of interaction goals. Frohlich's system is essentially designed as a system which enables *questions* to be posed by either dialogue participant, and thus contains elements of true interactivity, within a very limited interaction

style. Where a subject domain is conceptually rich, users may not be the best judges of what they need to know, and when and how they should be told it.

3.3 Argument and Dialogue

3.3.1 Reasoning, explanation, argument and dialogue

A relatively small body of research into computational generation and understanding of dialogue has focussed on generation and comprehension of *arguments*. We all have an intuitive idea about what an argument is, but the recent literature is not clear concerning the distinctions to be made between *reasoning*, *explanation* and *argument*. The work of Goguen, Weiner & Linde (1983) on 'natural explanation' adopts the following view on the distinction between reasoning and explanation:

"... reasoning is a phenomenon of the internal, mental world, which leaves no directly observable trace. ... an explanation is a unit of language which purports to show why the speaker believes some particular statement, and (in most cases) is intended to cause the hearer to accept this statement."

(Goguen, Weiner & Linde 1983, p. 523)

Their general approach follows that of Weiner (1980), described in the previous section of this chapter, where the relationship between reasoning and explanation is that an explanation is the 'output' of a reasoning process, expressed linguistically, and structured in a 'natural' way. There does seem to be general agreement on the notion that reasoning is primarily reserved for mental processes. For example, the work of Johnson-Laird (1983) on mental models, presupposes this view of reasoning in its persuasive argument against the idea of human reasoning as a kind of "mental logic". With respect to the notion of ('natural') explanation as textual expression of reasoning, there is clearly a sense in which a piece of text *can* be an 'explanation', just as an 'argument' may well be expressed by a single narrator in a book. The view expressed in the previous section of this chapter concerned with explanation was that *explanation in human-computer interactions*, is better viewed as something which *emerges* from an interaction, rather than being a specific form of interaction. What, therefore is *argument*, as a phenomenon in interactive dialogues, which distinguishes it from explanation dialogues? The answer seems to lie in the kinds of *dialogue structures* which may be commonly identified in each 'discourse genre'. In 'explanation' we find structures which express all and only the facts and rules which were used in reasoning to reach a conclusion, or a series of questions and answers designed to negotiate the mutual sharing and transfer of knowledge and belief. Arguments typically contain discussion of a variety of supporting and disconfirming evidence for a number of alternative conclusions which could be reached, where the function of argument dialogues seems to be that of *reasoning in dialogue*, rather than using dialogue to communicate and explain a reasoning process itself. Arguments therefore contain a number of characteristic 'units' (Alvarado, Dyer & Flowers 1986; Cohen 1987; Toulmin et al 1979), which are variously termed 'claims', 'warrants', 'justification', 'evidence', 'supports', 'backing' and so on.

Theories of argumentation therefore seem to diverge not on what the identifying dialogue structures of argument are, but on

- how these structures or elements are used in argument comprehension and generation ;
- how elements are typically grouped together ;
- what the function or 'goal' of argument is.

With respect to this latter point concerning the function of argument, Cohen (1987) states that

"... arguments have a defining characteristic - they are necessarily goal-oriented. The speaker wants to convince the hearer of some overall point."

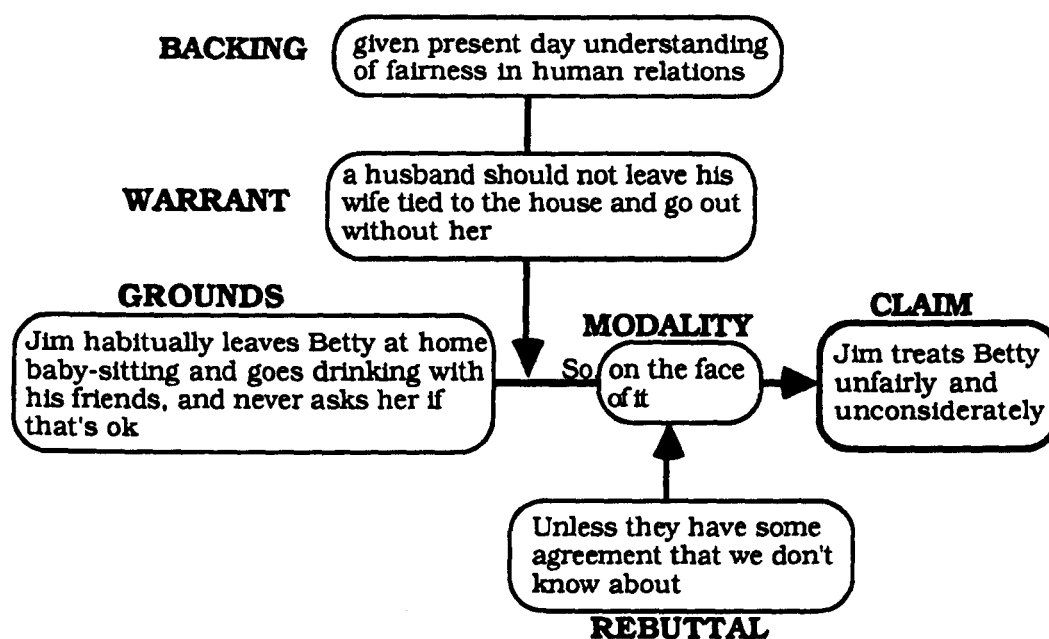
(Cohen 1987, p.11)

It seems acceptable that arguments are goal-oriented, since *all* discourse is in a sense goal-oriented provided the utterances were generated by speakers both consciously and intentionally. It is *not* however clear that the goal of argument must necessarily be that of 'convincing the hearer': it is possible that arguments could contain the dialogue units with which they are typically identified, whilst having different or more general goals. Arguments can be 'cooperative' rather than 'adversarial' in the sense that the cooperatively agreed goal between participants is generally to 'promote mutual belief revision', or to 'cooperatively arrive at some conclusion'. This notion of argument is embodied in the line of philosophical theories of 'dialectic', from Plato through to Hegel and Marx. It is possible that in the course of a speaker making some claim in an argument *their own* beliefs become revised. A special circumstance under which an argument may be cooperative is the case where one speaker is a 'teacher' and the other a 'student' - a situation which routinely obtains between teachers and students studying law, history, politics, physics, philosophy, and so on. In this situation a teacher may intermingle argument structures for claims, counterclaims, challenges and justifications with *explanation* of what those utterances mean in a domain which is not familiar to a student. This is the kind of argument which the intelligent tutoring system KANT is designed to support, described in the next two chapters. In the context of our present discussion, we can view the dialogues generated by KANT as combining elements of explanation, argument and instruction - an approach which moves away from the consideration of 'critical argument' as a separable discourse genre, towards a view of tutorial dialogues as dialogue *per se*.

There seems to be reasonable agreement on the kinds of linguistic structures which characterise argument as a genre. An influential approach has been the model of 'reasoned argument' developed by the philosopher Toulmin (Toulmin et al 1979). Arguments begin by some participant making a *claim*, which may then be given a set of *grounds* which support the claim. The necessity of establishing the relevance of grounds to claim is provided by *warrants*, which may themselves be given *backing*. Toulmin emphasises that *these elements do not appear to occur in any fixed order in human arguments*, and that we have therefore no fixed

'schema' for human arguments, but rather a set of elements which appear to be usually present in reasoned argument. Other common elements include *rebuttals* (often accompanied by the word 'unless ...') and *modalities*, which express a degree of confidence in the connection between grounds and claim. Figure 3.2 gives an example of such an argument analysed in this way.

Figure 3.2
An example argument according to Toulmin et al's (1979) argument analysis



This is not, however, the only type of argument type which has been identified. McGuigan & Black (1986) identify three basic argument types:

- (1) argument by analogy ;
- (2) categorical argument ;
- (3) causal argument.

Suppose, for example, someone makes the general statement "your dog will get fleas because mine did". This can be represented as

- (1) an analogical argument:

my dog has fleas
 you have a dog analogous to my dog, at the same level of generality of categories of animals
 so your dog is analogous to mine, and will get fleas

- (2) a deductive categorical argument:

For all x, if x is a dog, x gets fleas

your dog, d, isa dog
 therefore, by variable instantiation,
 d gets fleas

or an inductive categorical argument:

I have to date seen many instances of dogs with fleas, therefore I conclude that this new instance - your dog - will also get fleas

(3) the causal argument

"dogs like to run in the woods, making them accessible to fleas that jump at the opportunity to find a new home, and as a result of this your dog will get fleas"

(McGuigan & Black 1986, p. 242)

The important points about their analysis are the claim that there seems to be no privileged way of arguing, and their emphasis on the importance of the formation of general categories or concepts - such as that of the class of dogs. We shall see the influence of these approaches to argument element and type analysis in the brief review of some work on argument in the next section.

3.3.2 Approaches to generation and understanding of argument

One possible approach to computational generation and understanding of sustained argument is to specify some normative or *schematic* dialogue structure. A recent and complete exposition of this approach is Stutt's (1987) 'Arguing Expert System' (the Argument Support Program for Archaeology, ASPA), which builds on earlier work on argument structure by Alvarado, Dyer and Flowers (1986). The ASPA system uses a limited number of argument patterns or scripts (including 'reasoned argument' in Toulmin's terms, and argument by analogy) in an attempt to extend the explanation capabilities and user acceptance of second generation expert systems. As an interaction of argument exchanges proceeds, the system builds a dependency graph, based on Toulmin et al's (1979) analysis of sound human reasoning, in terms of 'grounds' for claims, where grounds have 'warrants' and 'backing'. One argument is assessed as being more 'convincing' than another in terms of the sum of certainty factors for individual claims, the 'plausibility' of the modal relation between grounds and claim, the pertinence of the claim to the current argument and the relevance of the evidence to the claim. The motivation behind using such argument scripts in expert system research is to give a greater symmetry between the operations on the knowledge base which are available to the user and system, thus improving user-acceptance. The argument exchanges may thus proceed in a way which lacks directedness, which is perfectly acceptable for the purposes for which the approach is intended. In a tutorial context, however, we need to take account of the goals of both the tutor and the learner, and integrate these goals within the overall dialogue. The kinds of activities which occur in critical arguments in a tutorial context include many features, such as informing, advising, correcting, and so on. We would therefore argue that for tutorial purposes, it would not be fruitful to consider the dialogue in terms of inflexible schematic structures, and that it may not be useful to

label the exchange as a 'critical argument'. The dialogue would certainly include elements of what we mean by argument, but should be able to flexibly accommodate the way in which a dialogue may alternate between criticism, explanation, statement, question, Socratic questioning and any other form of interaction which is viewed as achieving mutual goals of tutor and learner.

The MAGAC argument understanding and generation system developed by McGuigan & Black (1986) adopts a similar set of schematic structures for the argument types which they identify. As an argument is 'parsed', slots in frames are filled in for 'supports', 'categories' (in analogical arguments) and 'claims'. In addition to the use of schematic argument "units" in recognition of arguments, the work of Alvarado, Dyer & Flowers (1986) stressed the role of *domain specific knowledge* required in understanding an argument expressed as a piece of text. It seems clear that to understand arguments (and to *teach* with argument, as described in the next two chapters) knowledge, beliefs and their domain specific interrelationships need to be understood. As with Stutt's (1987) system, arguments are represented

"... in terms of an argument graph, which contains all propositions used by the argument participants. Propositions are connected by links that indicate how they support or attack one another. The argument graph aids understanding because the role of every new proposition is determined by establishing how the proposition can be integrated into the graph by using attack or support links."

(Alvarado, Dyer & Flowers 1986, p. 251)

The work of Cohen (1987) concentrates on argument understanding at a lower linguistic level, in terms of the knowledge and processes required by a passive listener to understand a speaker's argument. Input is restricted *a priori* to understanding arguments:

"Consider the task of designing ... a natural language understanding system (NLUS) where the input is restricted to arguments."

(Cohen 1987, p. 11)

Cohen states that an approach similar to using 'story schemas' cannot be used here : "... one is never sure what points the speaker will address; content can't be stereotyped" (Cohen 1987, p.11). In fact, the knowledge structure used in recognition turns out to be quite similar to the 'argument graph' used by Alvarado, Dyer & Flowers (1986) and Stutt (1987):

"The representation developed is a tree of claim and evidence relations comprising the argument, where a claim node is *father* to its evidence *sons*. In order to assign an interpretation for a given proposition, one must thus simply assign it a place in the tree. In this way, one can tell to which propositions it serves as evidence and from which other propositions it receives support."

(Cohen 1987, p. 12)

Assigning the function 'claim' or 'evidence' and their relations is done using identification of surface 'clue words'. The restrictions of this approach are clear, and most are stated by the author:

- the system is restricted to text which is *a priori* judged to be an argument ;
- it does not model argument as interactive dialogue, but rather the understanding by a single hearer ;
- within the first restriction, it is restricted to all and only the arguments which can be recognised as "... expected coherent forms of transmission...", in terms of clue words known to the system.

With respect to the issue of clue words, researchers studying a similar 'genre' - explanation - found that the surface structure of user's questions offered little clue as to the type or nature of explanation required (2nd Alvey Workshop on Explanation).

The approach of identifying schematic argument structures is an example of an approach to discourse analysis which Reichman (1985) terms that of identifying "discourse forms" (p.9). Her criticism of the "genre dictates structure" approach is as follows:

"Since it is clear that in any ordinary discourse or conversation, speakers use a number of different "genres" - stories, descriptions, arguments, for instance - what conventions govern the development of the whole conversation that includes these subparts? An adequate theory of discourse would need to specify the conventions or rules that apply across as well as within discourse forms..."

(Reichman 1985, p.9).

Given that we can isolate the elements of a discourse genre called 'argument' it seems that using a rigid schematic structure for generation and comprehension of arguments would have difficulty in doing justice to the flexibility with which these elements are combined in arguments, and the fact that in certain kinds of argument, features of other 'genres' may be freely included. For the purposes of generating tutorial dialogues based on argument, it may not in fact be profitable to characterise discourse genres in this way, given the varied goals and structures occurring in tutorial dialogues. In 'natural' dialogue, people seem to converse freely without restriction on the dialogue genre which is mutually understood to be used. We conclude that research into argumentation awaits integration within wider perspectives and theories of dialogue.

3.4 Computational Models of Dialogue

3.4.1 Introduction

In this section we shall review some areas of computational approaches to *dialogue theory*. The field can not be said to be a unified one, and relevant papers are to be found in journals and symposia concerned with cognitive science, artificial intelligence, philosophy and computational linguistics. We shall not be concerned with the separable field of 'discourse' in linguistics - other than the extent to which such work is referred to by computational approaches - nor with the field of 'conversation analysis', which derives from sociological and other empirical approaches to the study of naturally occurring conversations. Nevertheless, we can isolate a body of research which is concerned with computational (or 'artificial intelligence') modelling of *high-level dialogue*. Models of high-level dialogue are concerned with modelling the mental states of conversants, rather than the surface linguistic structures which they generate. Such mental states include the knowledge and beliefs of agents, their means of reasoning with and revising beliefs, and their conception of shared beliefs and goals. Kiss (1986) therefore views high-level dialogue as

"... dialogue which takes place between rational agents pursuing overlapping sets of goals..."
(Kiss 1986, p.1)

Implicit in this view of dialogue is the assumption that participants are essentially *cooperative* in their negotiation of the division and sharing of responsibility for shared goals. Related issues therefore concern modelling of agents' beliefs, and their means of reasoning with them and revising them - issues which we shall review in passing. Most of the literature concerned with cooperative high-level dialogue refers to a set of 'conversational maxims', described by the philosopher Paul Grice (Grice 1975). Grice's primary concerns were to distinguish between the kind of reasoning performed in formal logical systems, and the use of logical notions in human conversations. This led him to attempt to analyse the way in which logical *implication* is used in conversation, as "essentially connected with certain general features of discourse" (Grice 1975, p. 45). These general features of discourse were formulated in terms of a general 'cooperative principle':

"Make your conversational contribution such as is required, at the stage at which it occurs, by the accepted purpose or direction of the talk exchange in which you are engaged."
(Grice 1975, p. 45)

Echoing Immanuel Kant, a set of four categories of maxims were distinguished as part of the cooperative principle, concerning the QUANTITY of information provided, the QUALITY or intended truthfulness of information, RELATION, or appropriateness of the contribution, and the MANNER in which the contribution is said. For example, under the category of QUANTITY we have the ubiquitous maxim :

" 1. Make your contribution as informative as is required ..." (p. 45).

Under the category of QUALITY, we assume sincerity -

"Do not say what you believe to be false" -

under MANNER, conversants are expected to be clear and concise -

"Avoid obscurity of expression... Be brief ..." -

and under RELATION, conversants are expected to be to the point -

"Be relevant".

Most work in high-level dialogue has adopted - to a greater or lesser extent - the set of maxims suggested by Grice as a set of defining characteristics of the kind of cooperative dialogue which was being modelled. As well as providing a set of limiting assumptions for the interpretation of utterances in dialogue, they provide a set of implicit *normative* rules for 'good' or effective dialogue to be generated by a computational model.

In order to be 'relevant', speakers must have some conception of the 'topic' or focus, which is cooperatively understood in dialogue. Our review therefore discusses work which has concentrated on the modelling of focus or topic based units in dialogue. The 'manner' in which topics are expressed, is addressed by work on intentional or goal-based structures. Finally, we consider ways of modelling Grice's intuition that

" ... at each stage [in conversation] SOME possible conversational moves would be excluded as conversationally unsuitable."

(Grice 1975, p. 45)

The notion of generating utterances which are within this space of 'suitable conversational moves' has been variously modelled by grammars, schemata and planning systems. In chapters 4 and 5 we shall describe a different approach to generating a space of conversational moves, based on Grice's fundamental notion of *cooperativity*, which is achieved by *negotiation*.

3.4.2 Dialogue grammars and schemata

Grammatical approaches to modelling dialogue represent an attempt to transfer such concepts as 'well-formedness' and 'grammaticality' from the level of syntactic analysis at the sentence level, to the level of relationships between sentences, or super-sentential dialogue functional units. In its most simple form, this approach was literally confined to developing grammars for the relationships between sentences in written texts¹⁰. With respect to this approach, Levinson remarks that

"However adequate such a view may be for written non-dialogic text, it is simply not feasible as a model for conversation where the links between speakers cannot be paraphrased as sentential connectives..."

(Levinson 1983, p. 288)

¹⁰ See Levinson (1983), pp 287 - 289 for a critique of the 'text grammar' approach.

Grammatical approaches to dialogues - as opposed to text - often attempt to define a set of 'semantic categories' which are not necessarily identical to sentences, as the constituents of grammar rules. We have reviewed approaches of this kind in connection with explanation and argument, where such domain independent categories might be 'claim' or 'support'. Other approaches rely on essentially defining semantic categories of the *domain* of discourse - such as the semantic grammar embodied in SOPHIE (Brown & Burton 1982) and the explanation grammar for electronics of Cawsey (1988). Their restrictions are thus considerable - both to a restricted domain and discourse genre, if one can indeed be identified. DIAGRAM (Robinson 1982) is an attempt to produce a dialogue grammar beyond a specific discourse genre, but within a certain 'subset' of English. Examples of the grammar constituents identified are imperative, declarative and interrogative sentences. The model is based on a syntactic parser, which transfers such features as 'topic' identified by syntactic analysis of one dialogue level category to the interpretation of following categories. It does not deal with interactive conversations - being confined to the interpretation of utterances of a single speaker - and is not used for dialogue *generation*. It is in fact equivalent to a text grammar as described earlier, with all the attendant limitations of application to naturally occurring interactive dialogues. For any segment of apparently 'ill-formed' dialogue, it always seems possible to find a human context within which it seems intelligible and coherent (Levinson 1983). We may thus question the whole approach of applying sentence level concepts to interactive dialogue as being too restricted in application and flexibility. A case in point concerns the phenomenon of *adjacency pairs* (Schegloff & Sacks 1973), where a grammatical approach would have to predict a set of well-formedness rules for possible seconds to a first utterance. With respect to this point Levinson remarks that

"... what binds the parts of adjacency pairs together is not a formation rule of the sort that would specify that a question must receive an answer if it is to count as a well-formed discourse, but the setting up of specific expectations which have to be attended to."
(Levinson 1983, p. 306)

Grammatical theories attempt to account for both the *goal (intention)* and *topic* of utterances which may well-formedly follow others. These notions are grouped together in the discourse grammar developed by Reichman (1985), which identifies a set of "context spaces" as the fundamental dialogue units, and a set of grammar rules for transitions between them. Since context spaces are based on a theory of utterances grouped together primarily in terms of their topic, we shall discuss her theory in more detail in the next section. At this point in our review it is worth remarking that since her theory is based on a notion of *grammar* (implemented as an augmented transition network) as the connection between dialogue units, many of the above criticisms apply to this feature of the theory.

In the development of research in computational linguistics, the problem that syntactically ill-formed utterances - such as "thesis writing tired" - could nevertheless be effective

communications in a given context, led to an emphasis on matching sentences with a representation of their *semantics* or 'meaning'. Typical examples included the use of frames (Minsky 1977) and scripts (Schank & Abelson 1977) in understanding written stories. A similar approach was used in the GUS system ("a frame driven dialog system") for generating dialogues in the restricted domain of air travel enquiries (Bobrow, Kaplan, Kay, Norman, Thompson & Winograd 1977). GUS used a set of frames at several levels, where the utterances generated by the system were those required to fill in slot values in frames for the stereotypical pattern of a dialogue within its domain. The system begins the dialogue by creating an instance frame for the dialogue, and generating a *question* using a 'demon' procedure ('ASKCLIENT') attached to the first slot. The client answer "I want to go to San Diego.", to the question "Where do you want to go?", is parsed using case frames for the verb 'to go', creating a subframe with slots to be filled in for 'agent', 'to-place' and 'date', the filling in of which is put on the agenda of tasks to perform. Fundamentally, the dialogue is driven by the successive filling in of frame slots by either attached 'demon' procedures which are activated automatically when a slot receives a value, 'servants', which are activated on demand from the central control mechanism or default values which are given in the absence of input information. The dialogue is 'mixed-initiative' in the sense that the user can also ask questions, which are interpreted using the same syntactic and case frame analysers. The authors summarised the approach as follows:

"GUS attempts to control a conversation by fitting it to the mold laid down in a structure of related frames. It has a place prepared in this structure for each piece of information that might potentially be used for making travel arrangements. It also has a strategy that will cause the pieces of information that the client must supply to be elicited in a natural order. The sequence of slots in the frames determines the usual course of the conversation, but it will change if, for example, the client volunteers information or asks questions."

(Bobrow, Kaplan, Kay, Norman, Thompson & Winograd 1977, p. 170)

In other words, execution of the task agenda is interrupted by user questions, some tasks may be popped off the stack by user volunteered information, and the agenda is resumed after the interruption. GUS exhibits the strengths and limitations of domain restriction of a dialogue system. It works well "... thanks largely to the very narrow expectations it has about the subject matter and the client's goals" (op cit p. 157). Operation within less restricted domains would require a large number of frames for "conversational patterns" and the attendant problem of deciding which to use in a given context. The major theoretical problem lies in the identification of 'conversational patterns' which are in fact completely determined by the structure of the *domain*: although task-related dialogues do have a structure which mirrors the task itself (Grosz 1977), it is clear that there are other perspectives on dialogue structure which are functional or goal-based, and thus relatively domain independent. An adequate theory of dialogue must do justice to both.

3.4.3 Topic structures in dialogue

In the previous section we discussed models for dialogue which were based on identification of a set of units of dialogue in terms of semantic primitives or domain schemata, and a set of grammar rules or frames which expressed their interrelation. In this section we discuss computational models for dialogue which assign a predominant role to dialogue units based on *topic* or *focus*. Topic-based approaches include the work of Grosz (1977), which has been extended by Reichman (1985). Intention based approaches view dialogue as fundamentally a mechanism which conversants use to achieve their goals, and argue that these goals can be achieved by an *integration* of physical and dialogue actions (Power 1979, for example). All of these theories share the set of very general assumptions of the theoretical approach which Levinson (1983) terms 'discourse analysis', which are

"(a) the isolation of a set of basic categories or units of discourse, (b) the formulation of a set of concatenation rules stated over those categories ..."

(Levinson 1983, p. 286)

In reviewing these approaches we shall therefore aim to state clearly what the fundamental units of discourse are claimed to be, and how they are claimed to be interrelated.

3.4.3.1 Focus and task-oriented dialogues

The work of Grosz (1977) concentrated on dialogue units based on the domain of discourse, and how a conversant's conception of *focus* shifts as the dialogue progresses. The theory advanced is restricted to dialogues concerned with describing some *task* (such as instructing someone how to bolt a metal plate onto something):

"The shift strategy [i.e. of focus] described here is specific to task-oriented dialogs. It reflects the tasks as the major topic of such dialogs, and hence the major indicator of shifts of focus. Although the rest of the focus representation is general, this aspect would need modification for application to other kinds of discourse."

(Grosz 1977, p. 71)

Focus is fundamentally based on *partitioning* (Hendrix 1975) of a semantic network representation¹¹ for the task-domain knowledge ("... the network is partitioned to encode focus ..."; Grosz 1977, p. 68). Items in *explicit focus* are those which have been explicitly mentioned in the preceding dialogue, together with a *type* relation for each item. For example, if the node 'book' is in explicit focus, and we previously mentioned 'owning' rather than 'in library', then it is the former rather than the latter relation which is also brought into explicit focus. Items in *implicit focus* are "... those items that are relevant because they are closely connected to items in explicit focus" (op cit, p. 68). A further distinction is made between *active*, *open* and *closed* focus spaces:

¹¹ Hendrix's work was in fact initially developed in the context of Intelligent Tutoring Systems research.

"At any point in a dialog, only one focus space is "active," but several may be considered "open". The active focus space reflects the focus of attention at the current point in the dialog. The open focus spaces reflect previous active spaces that contain some unfinished topics and hence may become active again ; they are areas to which the dialog may return."

(op cit, p. 69)

A further theoretical construct introduced by Grosz is that of a *hierarchy* of context spaces. This relates to Hendrix's (1975) notion of partitioning semantic networks into 'vistas', and reflects the particular 'point of view' given to a concept, in terms of the type of relations in which it participates, as described earlier. For example, 'bolts' in Grosz's example task domain, can be viewed in terms of the 'EXCHANGE' relation, and other nodes attached to this, or in terms of the relation 'ATTACHING', and its related nodes. Explicit focus therefore relates also to isolating one of these vistas as the current "orthodox vista": *the hierarchy of context spaces is simply determined by the hierarchy of tasks and subtasks in the given domain.* Finally, after identifying explicit, implicit, open and closed *context spaces* as the fundamental dialogue units, Grosz postulates a simple mechanism for their succession in task-oriented dialogues: "In task dialogs , a shift in focus takes place whenever a new task is entered or an old one completed" (op cit, p. 72). In other words, when a task and its subtasks is completed, all open context spaces will be closed, and the focus shifts, to an item in *implicit focus* :

"...references to implicitly focused items are considered indications of shifts of focus."

(op cit, p. 68).

Shifting from a context space concerned with a task to one relating to its subtasks does not constitute a focus shift, since if discussion of the high-level task was not completed, its space remains *open*, to be *returned to* after discussion of subtasks in the hierarchy:

"A narrowing of focus takes place whenever a subtasks of the active task is opened for discussion. The focus shifts back up to the higher level task when that subtask is completed. Hence, when a subtask of the current task is referenced, a new active focus space is created below the current active focus space. When the subtask is completed, the new focus space is closed and the old space (i.e. the higher space) become the active focus space again. The top of the focus space hierarchy is the focus of the overall task."

(op cit, p. 72)

We have discussed Grosz's work in some detail because the concepts of 'focus spaces' (active, open, closed) and explicit and implicit focus have been influential in the literature. The limitations of the approach to *task-based dialogues* and to the problems of *focus* alone were clearly stated by Grosz.

3.4.3.2 Context spaces and conversational moves

The work of Reichman (1985) was originally based on that of Grosz (1977), and concentrates on the definition of "context spaces" and the relationships between them. Reichman's notion of a 'context space' is much wider than Grosz's notion of a 'focus space', on which it was originally based: context spaces are not intended to be essentially limited to a *domain* (task or otherwise), and are claimed to be a fundamental set of dialogue structures identified from analysis of naturally occurring conversations. Fundamentally, context spaces are *states* in an augmented transition network (see Reichman 1985, pp. 98-99 for examples), with state transitions controlled by a *grammar* of "conversational moves". Before discussing these 'moves', let us try to describe the complex theoretical definition of a "context space".

The theory is extremely complex and is argued for in great detail. At the most general level, a distinction is made between "issue" and "non-issue"¹² context spaces [henceforth 'cs']. In an *issue cs*

"... the truth, necessity, goodness or appropriateness of a particular state of affairs is discussed"

(Reichman 1985, p. 56)

with appropriate subtypes for "evaluative", "epistemic" and "deontic" cs's. *Non-issue* cs's are concerned with "... comment, narrative and two types of support ..." (Reichman 1985, p. 58). All context spaces have a standard set of slots (with special extra slots defining subtypes) for the following:

type - category name

derivation - whether claims explicitly stated or inferred by system

goal - the 'conversational move' performed by the cs - eg 'support'

contextual function - method used to achieve goal

speakers - persons who generated utterances lying in cs

status - foreground or background role

focus - high, medium, low or zero

We can therefore note that, in comparison with Grosz's approach, for Reichman focus is a matter of *degree*, and she adopts a similar hierarchical relationship between spaces and subspaces. In addition, it seems that the primary way of viewing a cs in in terms of a group of utterances which perform a particular *conversational move*: the *type* of cs is primarily defined as the *conversational move* which it aims to achieve (for example, a "non-narrative support" cs has the goal of performing a "support" conversational move - Reichman 1985, p. 61). We turn therefore to the definition of a conversational move. A conversational move comprises a *functional unit* in dialogue, which has a certain structure of expected subcomponents, a set of 'clue words' which typically accompany the set of utterances comprising the move, and a set of

¹²In earlier papers Reichman (1978) had used the term 'events' for non-issue spaces.

effects on the status of the context space established by that move. An example is the 'SUPPORT' conversational move:

move: SUPPORT

components:

claim

supports

principle of support

mappings (between claim and principle)

clue words:

"Because...", "Like ...", "When ..."

effects:

if preceding active context space is SUPPORT, closes previous cs; if ISSUE, new cs is in controlling status

(simplified from Reichman 1985, p. 37)

The components of the move mean that each move is a kind of "schema" (Reichman 1985, p. 37). How, then, are conversational moves and context spaces used in dialogue generation and understanding? Context spaces and their associated conversational moves are connected together into an augmented transition network, with tests on each arc which place restrictions on transitions from the current state (cs/conversational move) to other possible states. Reichman also states that

"Traversal of the discourse ATN entails the production or parsing of a single conversational move. Arc tests specify the developmental options available at any point in a discourse - what moves are viable in the given discourse context. A complete discourse is facilitated by many cycles through the different network paths."

(Reichman 1985, p. 93).

This would seem to imply that several ATNs - or a set of connected ATNs? - are required for each conversational move, but we are given no clear indication as to how the model chooses which possible move to follow in discourse generation.

In general, we could say that although the model appears to identify a number of recurring features in natural conversations, it is both *incomplete* and *inconsistent* (see the discussion Reichman 1985, pp. 93-97). The title of Reichman's book derived from her thesis - "Getting computers to Talk Like You and Me" - therefore seems to make far too strong a claim for the theory. At certain points in the discussion it appears that state transitions are effected by arc tests, and at others they are part of the conversational move itself. Also, arcs do not just perform tests to determine which state/move to go to next: they also "... generate the clue words associated with a given conversational move and the substantive utterances filling its mode of development." (Reichman 1985, p. 96). At some points Reichman suggests that the model is one

for generation *and* understanding of dialogue as an interactive process between a computer and a human user, and at other points that the computer performs both roles within the model, simulating the user. Considerable extensions would be required of the theory in order to do this. Many of these problems appear to arise from the fact that

"... in our system transition tests really correspond to calls on other sophisticated subsystems of a full computerized natural language system. Such subsystems of course do not currently exist, though they are the center of much current research effort."

(Reichman 1985, p. 95)

A different criticism concerns the *generality* of Reichman's theory, since she claims to have identified a set of rules which apply to any sort of discourse "regardless of its purpose and subject matter, and regardless of the situational context"¹³. Delin (1986) has attempted to apply Reichman's rules to other forms of dialogue and expressed considerable doubts concerning their universality.

If we consider the notion of the "context space" as Reichman's fundamental unit of discourse, it is clear that it is called upon to fulfill a great many functions: it attempts to capture notions of both topical and functional or goal-based units within a single representation. We would argue that a simple augmented transition network provides insufficient formal tools to bear the weight of her analysis. Grosz makes a similar point about Reichman's extension of the original notion of 'focus space' to that of 'context space':

"Several researchers (e.g. Linde & Goguen, Reichman-Adar 1984) misinterpreted the original research in an unfortunate and unintended way: they took the focus-space hierarchy to include (or to be identical to) the task structure. The conflation of these two structures forces a single structure to contain information about attentional state, intentional relationships, and general task knowledge. It prevents a theory from accounting adequately for certain aspects of discourse, including interruptions ..."

(Grosz & Sidner 1986, p. 182)

It seems that a theory of dialogue must do justice to - at least - topic based and goal based dialogue structures, but that the conflation of the two into a single representation (as in Reichman's theory) gives us little clear theoretical understanding of the complex relationship between the two. The later work of Grosz (Grosz & Sidner 1986) attempted to specify just such a relationship, in combination with the work on parsing of *intentions* in discourse performed by Sidner (1985).

¹³Quoted in the the review by Delin (1986).

3.4.4 Topic and Intention in Dialogue

We shall discuss Sidner's (1985) work on intention recognition at this point (rather than in the next section on goals and planning) since it was integrated with Grosz's theory, described earlier, to provide an integrated model of intention and 'attention' in discourse. Sidner's (1985) work addressed the problem of understanding discourse by considering the subproblem of how a hearer infers a speaker's intentions and plans in discourse:

"One theoretical problem to be addressed by natural language research is how the hearer delineates a discourse into units and then in each unit recognizes the speaker's intention for the hearer. In addition, the problem of how the hearer uses the speaker's intention to infer an intended response must be addressed."

(Sidner 1985 p. 1)

The proposed approach is simple in theory and detailed in actuality:

"To determine an intended response, the hearer must ... First ... determine which utterances in a discourse form a unit in which information about the speaker's intention is expressed. Second, the hearer must extract the speaker's intention from the information in the unit. Third, the hearer must create a description of what his intended response will be."

(Sidner 1985, p.1)

Speaker's are claimed to recognise intentional units from surface "... markers for the beginning and end of units." (op cit, p.2), where a unit is defined as "... a portion of a discourse in which all the utterances provide information about one general intended response." (p.2). For example, in the 'discourse'

"I want you to sing me a song.

It's "Yankee Doodle Dandy" in the key of C."

there are two surface *intentions* (1. to sing "Yankee Doodle", 2. to sing it in C), there is only one *intended response* (the hearer's singing Yankee Doodle in C). The hearer's recognition of the speaker's intention proceeds basically by (partial) matching of the inferred intended responses from the speaker's utterances with a predefined space of *potential speaker plans*. Sidner recognizes that this restriction is "too strong" for discourse, and further states that there is a major problem in intention recognition, since the key utterances which relate directly to intention may occur at unpredictable points in the set of utterances which comprise an initial speaker's statement. Her work thus relates directly to that of Grosz (1977) when she states in conclusion that in order to pursue this and other issues "... researchers will have to address questions about the role of focus (Grosz 1979)" (Sidner 1985, p. 9).

3.4.4.1 Topic, Intention and Attention in Dialogue

The key problem of producing a theory which integrates functional and focus-based theories of dialogue was addressed in Grosz & Sidner, 1986 - a key paper in the literature on computational models of dialogue, which merits close analysis. It consists of a theory which is very simple in its general form, which is argued in impressive detail. The theory is primarily *descriptive* of discourse structures, which are said to consist of three separate but interrelated components:

"Our main thesis is that the structure of any discourse is a composite of three distinct but interactive components:

- the structure of the actual sequence of utterances in the discourse;
- a structure of intentions;
- an attentional state."

(Grosz & Sidner 1986, p. 176)

Linguistic structure consists of the way in which the actual words and phrases of discourse "... are naturally aggregated into discourse segments." (p. 177). It is primarily indicated by pauses, and linguistic 'clue phrases', together with an "embedding relationship" between segments, which is primarily a reflection of an embedding of *intentional structure*:

"The linguistic structure consists of the discourse segments and an embedding relationship that can hold between them... the embedding relationships are a surface reflection of relationships among elements of the intentional structure."

(Grosz & Sidner 1986, p.177)

Intentional structure is the structure of goals or purposes¹⁴, which are restricted to those purposes which are intended to be *recognised*, as in the earlier work of Sidner (1985). Given the premise of linguistic segmentation, they adopt the initial simplification of assuming that for any discourse there is one *discourse purpose*, and that for each discourse segment "... we can also single out one intention - the discourse segment purpose ..." (p. 178). A discourse participant may engage in a discourse for purposes which remain private, but "... the discourse segment purpose is always intended to be recognized." (p. 178). The hierarchy of intentional structure is expressed in two *structural relations* which obtain between discourse purposes ('dsp's):

if dsp₁ must be satisfied before dsp₂, dsp₁ satisfaction-precedes dsp₂

if dsp₁ provides part of the satisfaction of dsp₂, dsp₂ dominates dsp₁

These relations place a crucial part in recognition of intentional structure in discourse. Since it is clear that for any segment, the range of possible intentions which it could relate to is

¹⁴ For the present, in reviewing existing work, we have used the terms 'goal', 'purpose' and 'intention' interchangeably, as have the authors reviewed. In the next section we describe the work of Cohen & Levesque (1987) which distinguishes intentions from their embedding as goals in planning systems.

intrinsically open-ended, thus "... a theory of discourse structure cannot depend on choosing the DP/DSP from a fixed list..." (p. 179): "Since the CPs ['conversational participants] can never know the whole set of intentions that might serve as DP/DSPs, what they must recognize is the relevant structural relationships among intentions." (p. 179).

Attentional state - the third component of discourse structure - is "... an abstraction of the participants' focus of attention as their discourse unfolds." (p. 179). As with Grosz's earlier work (1977), it is modelled by a set of *focus spaces*, where one focus space is associated with each discourse segment, and changes in attentional state are modelled by "... a set of transition rules that specify the conditions for adding and deleting spaces." (p. 179). If each discourse segment has a discourse purpose and a focus space associated with it, the crucial question concerns the relationship between intentional and attentional structures, and how changes in each effect each other. The first thing to note is that in the general process of 'focussing', conversants "... are focussed not only on what they are talking about, but also on why they are talking about it." (p. 180). Each focus space therefore "... also includes the DSP ..." (p. 179), as, incidentally, in the work of Reichman (1985). Within attentional structures, focus spaces are embedded in the form of a stack; and within intentional structures, we have the relations of dominance and satisfaction-precedence. The key to their interrelation lies in a generalisation of Grosz's earlier claims that attentional structure mirrors *task structure*, but is conceptually separate from it. *Intentional structure* now fulfills the same theoretical role as *task structure*. In the earlier work, "... a shift in focus takes place whenever a new task is entered or an old one completed" (Grosz 1977, p. 72). If we substitute 'task' for 'intention', then we find that

"... the focusing structure is parasitic upon the intentional structure, in the sense that the relationships amongs DSPs determine pushes and pops. [i.e. in focus spaces - my comment]" (Grosz & Sidner 1986, p. 180)

We may now summarise their theory. For discourse generation, the implications are quite simple: if a discourse generation mechanism has a current goal in terms of utterance type and topic, then goal shifts - as part of plan execution or otherwise - are the primary decisions to be made, which have topic shifts as side-effects. The decision process takes place within the *constraints* of a set of rules for 'coherent' topic shifts, and established dominance (i.e. subgoal) and satisfaction-precedence (i.e. precondition) relationships between intentions (i.e. goals). In other words, changes in dialogue goals are constrained by *local* focus constraints:

"A primary role of the focus space stack is to constrain the range of DSPs considered as candidates for domination or satisfaction-precedence of the DSP of the current segment. Only those DSPs in some space on the focusing stack are viable prospects. As a result of this use of the focusing structure, the theory predicts that this decision will be a local one with respect to attentional state ... this prediction corresponds to a claim that locality in the focusing structure is what matters to determination of the intentional structure. "

(Grosz & Sidner 1986, p. 191)

There are two final and extremely important corollaries to their theory which need to be mentioned. The first concerns the notion of "topic" in sentences and in discourse, where they make the "intriguing conjecture" that

"... "topic" is a concept that is used ambiguously for both the DSP of a segment and the center" (Grosz & Sidner 1986, p. 191)

where the 'center' is the referent in a previous discourse segment which is used to constrain search for pronoun referents in the present segment, and which may shift within the segment. In addition, they state that

"It appears that many of the descriptions of sentence topic correspond (though not always) to centers, while discourse topic corresponds to the DSP of a segment or of the discourse." (p. 192)

The second point to note concerns the way in which changes in attentional and intentional state are communicated in discourse, and relate to the "cue words" used:

"ICPs rarely change attention by directly and explicitly referring to attentional state ... Likewise, discourses only occasionally include an explicit reference to a change in purpose... More typically, ICPs employ indirect means of indicating that a change is coming and what kind of change it is. Cue phrases provide abbreviated, indirect means of indicating these changes." (Grosz & Sidner 1986, p. 196).

This relates to the very important issue of *implicit negotiation* in cooperative dialogues, addressed by the work of Levin & Moore (1977) and discussed in the following chapters 4 and 5. Unlike Grosz & Sidner, Levin and Moore's theory does not depend on identification of surface linguistic markers, but rather on the identification of referents in dialogue which may be interpreted as 'parameters' of a proposed intentional structure.

It is reasonably clear how the theory advanced by Grosz and Sidner relates to discourse generation, but a number of unsolved problems remain in the problem of a conversational participant's *recognition* of the structures and their interrelationships postulated by the theory. A problem that they recognise is the assumption that a *single* discourse purpose correlates with a single discourse segment:

"The assumption that there are single such intentions will in the end prove too strong. ... We must leave to future research ... the exploration and discussion of the complications that result from relaxing this assumption." (Grosz & Sidner 1986, p. 178)

Is this assumption simply a temporary assumption which allows the authors "... to describe the basic theory more clearly." (p. 178), or is it a problem which renders the theory seriously flawed in principle? Levinson (1983) would argue the latter. His criticisms concern the assumption of speech act (Searle 1969) theories that a single speech act can be correlated with each discourse segment, but the same applies to correlating single 'intentions' with single discourse segments. The basic idea of approaches such as that of Grosz & Sidner is that since the observable regularities in discourse - such as 'offers are followed by acceptance or rejection' - can take an unlimited variety of forms at the linguistic level, if we can perform some translation into more abstract units, then the interrelationships will be more easily specifiable at this level. A detailed consideration of these issues in philosophy of language and linguistics is somewhat beyond the scope of our review here. But we may remark that since any given utterance could serve an unrestricted number of intentions (and single utterances can achieve multiple goals), we have no *a priori* way of knowing whether relationships between abstract units at this level might not be just as complex as those observed at the linguistic level, even supposing that the identification of a range of possible intentional dialogue units was unproblematic. Levinson would claim that "... the attempts to bridge the gap (between what utterances 'literally' mean and 'actually' do in the way of actions) with theories of indirect speech acts have provided at best only partial solutions." (Levinson 1983, p. 291). If instead of attempting to produce a general model of 'naturally occurring human dialogues', such models are restricted to *human-computer dialogues*, then the simplifying restrictions become more feasible, since it may be reasonable to assume that the user's plans, and intentions are restricted to those which are possible within a computer-based system. This will *not*, however, be a reasonable assumption where a user is not aware of the range of possible operations in a computer-based environment, and one of the primary functions of a dialogue model is to infer the user's intentions.

3.4.4.2 Dialogue games

Levin and Moore (1977) proposed a model of dialogue which is fundamentally based on the identification of goal-based or intentional dialogue units - called "dialogue games" - but which makes some general statement about their relationship with topic shifts. We mentioned the model briefly in discussing the intelligent tutoring system IMPART (Elsom-Cook 1984; section 3.1 of this chapter), and its use of dialogue games for discussing topics or "issues" to discuss in a guided-discovery environment for the programming language Lisp. In Levin and Moore's theory, these functional units have primacy over focus-based units: shifts of topic (or focus) are dependent on changes in the goal structures of conversants, and are accomplished as side effects of goal structure changes. Dialogue games may span several sentence units and can not be identified with turn-taking in dialogue. Once a dialogue game is implicitly recognised to be in operation by participants, it is used in their *comprehension* of utterances within the context of that functional unit. For example, one functional unit (they identify six) is INFORMATION-SEEK. Once it is recognised by speakers that this is in operation, it gives the context for interpreting utterances to seek and supply information. There will be a set of dialogue rules for

information requests and a different set for supplying information.

In all dialogue games, person 1 wants something and initiates the dialogue in an attempt to achieve that goal. Dialogue games have a structure of three parts: (1) parameters, (2) parameter specifications, and (3) components. For example, the dialogue game INFO-SEEK has the following structure:

parameters

seeker

source

info

parameter specifications

s1: not(knows(seeker info)).

s2: knows(source info).

s3: wants(seeker s1).

s4: willing(source s1).

components

wants(source aware(seeker info)).

wants(seeker aware(source obligated-to(seeker source))).

(simplified from Levin & Moore 1977, p.408)

Dialogue games may be nested within each other in dialogue, and one dialogue game may be used as part of the process of negotiating which dialogue game should be pursued. For example, the HELPING dialogue game may be preceded by an INFO-SEEK dialogue game in which one speaker tries to get information from the other about whether they possess the knowledge required in order to be able to help them. It should be emphasised that the model proposed by Levin and Moore is one for *comprehension* by a single dialogue participant.

Let us consider the processes involved in using dialogue games in comprehension in more detail. The comprehension model includes a long-term memory containing domain specific knowledge, a representation of the knowledge of the other dialogue participant and a set of dialogue games, a 'workspace', or working memory and a set of processors that modify working memory. Five stages of processing are involved: (1) nomination, (2) recognition, (3) instantiation, (4) conduct, (5) termination.

nomination

This is the process of 'nominating' one or more possible dialogue games to be used in recognition by analysis of "... attempts to establish various entities as the values of the Parameters of the desired dialogue-game" (p. 411). In fact, there are two ways in which this is done. The first, concerns the *parameters themselves*, of which there are only three: "... the two participants involved ... and the subject of the dialogue (called "Topic")."(p. 402), where a 'spreading activation' model of memory (based on Collins & Loftus's (1975) work in this area) brings new

possible concepts into working memory, which could fulfill the role of the topic parameter. The second concerns the *parameter specifications* - eg (helped wants (helped performs task)) for the HELPING dialogue game - where (indirect) references to 'wanting', for example, could match with the specifications of a number of dialogue games, thus also nominating them as candidate dialogue games.

recognition

It is clear from the nomination process, that a number of possible concepts could become activated and a number of possible parameter specifications could be matched by a single initial statement - for example, all of the games INFORMATION-SEEK, INFORMATION-PROBE, INSTRUCTING, and so on contain requests for information. In the recognition process, partial matching and 'plausible inference' methods are used to try to eliminate candidates by checking whether a game has one or more specifications which are mutually contradictory, and preferring those with most supporting evidence. For example, suppose the question "How do I get RUNOFF to work?" leads to nomination of both INFO-SEEK and INFO-PROBE dialogue games (as described earlier). They differ in the respect that INFO-SEEK has a parameter specification which says that the questioner does *not* know the answer, and in INFO-PROBE, the questioner *does*. Looking at the listener's model of the speaker's knowledge should help to decide between the two alternatives.

instantiation

At this stage it is assumed that previous processes have filtered out certain games, and that a single game remains to be *instantiated*. The parameter specifications of the chosen dialogue game are placed into working memory "... as new inferred knowledge about the Parameters." (p. 413). Levin and Moore claimed that dialogue games "... can form the basis for deriving the indirect effect..." (p. 399) within a dialogue segment. In simple terms, the new predicates for parameter specifications of the selected dialogue game are held to encapsulate the *indirect effect* of the utterance from which their attendant dialogue game was inferred. For example, suppose the HELPING dialogue game is instantiated. We therefore have predicates such as

SPEAKER believes HEARER knows how to do task.

Supposing that HELPING was instantiated from the original utterance "I can't get this thing to work", then this is taken to be a request for help (which on the surface it is not).

conduct

"Once a Dialogue-Game is instantiated, the Dialogue-Games Processor is guided by its components in comprehending the rest of the dialogue. For the speaker, these goals guide what he is next to say; for the hearer, these provide expectations for the functions to be served by the speaker's next utterances." For example, in the HELPING dialogue game, the HELPER will utter the help specified in 'enabling' and the helped will perform the 'task' in the parameter specification

(HELPER ABLE (HELPER ENABLE (HELPEE PERFORMS TASK)))

termination

A dialogue game is terminated when one or more of its parameter specifications no longer holds as true. Presumably a specification can become 'untrue' when the goal of the game is achieved or if someone changes their goal - for example the HELPING dialogue game could become untrue if help has been given and so is no longer wanted, or if someone no longer agrees to help.

The work of Levin & Moore has been influential in dialogue research, so it is worth trying to summarise its main features, in order to clarify which aspects of dialogue research the work addresses. The theory models generation and understanding *for a single dialogue participant*. We are given no information on how the model could be extended to generate interactive multiparty dialogues, including mechanisms for organising turn-taking and the like. We are given a theory which describes how a bid to pursue a particular dialogue game is *interpreted* and subsequently *conducted*, but it should be made clear that no high-level goal structure is described for showing how a dialogue participant would decide which games to *nominate*. The description of the nomination phase would in fact contribute to a theory of *implicit negotiation* of dialogue games - a form of "indirect speech" - in interactive dialogue (see later discussion of the work of Power 1979). The authors claim to have produced a theory which addresses and solves many of the problems of 'indirect speech'. They give the example of the statement

"I can't get this thing to work" (p. 413)

which is recognised as a piece of indirect speech for requesting *help* - i.e. nominating the HELPING dialogue game. The plausible matching techniques can work reasonably well because we are considering a restricted domain with a restricted number of possible *intentions*. The work of Sidner (1985) and Grosz and Sidner (1986) - discussed earlier in this section - argued that the 'plausible' assumption of a single intention (i.e. dialogue game) for an utterance was not, however, plausible for unrestricted human-human dialogues. Consider the statement "It's late, isn't it". This could be given any number of direct and indirect interpretations depending on the context and a large amount of real world knowledge and knowledge of the speaker's beliefs, knowledge, goals and desires. Levin and Moore do not describe how this knowledge can be modelled, which must be crucial to the success of their plausible reasoning techniques for matching intentions as dialogue games. We therefore do not necessarily have a model of how intentions are interpreted in human dialogues.

Finally, we need to clarify *the relationship between intention and topic* implicit in Levin and Moore's theory. They state that

"Changes of 'topic' in dialogue are directly dependent upon changes in the participants' goal structures, and are accomplished as side effects of goal structure changes" (p. 400)

"Goal structure changes" must be coincident with the *nomination* of a new dialogue game. However, if the spreading activation model for new topics which they describe acts in the mind of the hearer in *interpretation*, then this activation process also seems to act to partially match possible dialogue games to pursue in the mind of the speaker. In other words, topic changes nominate possible intentions through partial matching. We may say therefore, that the interrelationship between topic and intention is stated in their theory, but is not specified clearly and consistently.

We have reviewed the theory of Levin and Moore in detail since it is probably the most closely related research to the model for generating tutorial dialogues described in chapter 5 of this thesis. In a manner similar to Levin and Moore, we use a set of intentional structures called 'dialogue moves', and local topic constraints based on a spreading activation model (Anderson 1983). In chapter 6 we perform a detailed critique of the two approaches, after we have fully described the KANT system.

3.4.4.3 Summary: Topic and Intention in Dialogue

This section has been concerned with critically reviewing approaches to characterising dialogue focus, or "topic", and its relationship with function, goals or intentions in dialogue. The approaches may be summarised as follows:

- Grosz (1977) described how dialogue focus could be analysed as a hierarchical set of "focus spaces", whose structure mirrored that of the *task* being discussed.
- Reichman (1985) generalised the notion of 'focus space' to that of "context space", being a single representational structure in dialogue which combined topical structures with associated functional structures called "conversational moves". Transitions between context spaces were described by a *grammar*.
- Grosz & Sidner (1986) produced a theory which generalised the focus-space/task structure relationship to a relationship between topic-based structures ("attentional state") and intention-based structures ("intentional state"). Changes in attentional state (goal structure) cause changes in attentional state within the local constraints of the latter.
- Levin & Moore (1977) concentrated on intentional or goal based dialogue units called "dialogue games". Topic shifts occur as side-effects of dialogue game changes via a dialogue game *parameter* for its topic, and spreading activation in turn suggests a space of possible dialogue games to pursue via partial matching with dialogue game parameters.

It seems, therefore, that a crucial question for research into computational modelling of dialogue concerns *the relationship between topic and intention*. Most authors agree (*contra* Reichman, 1985) that they should be given theoretically separable but interacting structures, but that dialogues are primarily defined in terms of their goal or *function*, within local topic constraints.

3.4.5 Dialogue, Goals and Planning

The approaches to integrating topical and intentional structures described in the previous section stressed the importance of "A Goal-Oriented View of Language" (Levin & Moore 1977, p. 415). Thus Levin & Moore (1977) describe the following as a 'meta-goal of comprehension':

"To comprehend an utterance, find some previously known goal of the speaker which this utterance can be seen as furthering" (p. 415),

Sidner (1985) states that in order for a hearer to recognise what some speaker wants them to do

"... the hearer must ascertain what plans the speaker is undertaking and how the utterances in the discourse further that plan."

(Sidner 1985, p.1).

All of these authors agree that goal shifts effect topic shifts within certain constraints: they do not describe how these goal shifts take place, nor do they describe theories of dialogue generation which are reasonably *complete*. The theories reviewed in this section concentrate on the analysis and representation of goal-based dialogue units, and the way in which speakers combine goals or intentions into sequences, commonly as *plans*.

3.4.5.1 Planning Cooperative Dialogues

The work of Power (1979) adopted a view of dialogue where actions to achieve goals can be either physical or linguistic:

"Linguistic acts belong to the repertory of methods by which we achieve our goals."

(Power 1979, p. 107).

His work was concerned with modelling dialogue generation between two hypothetical conversants who cooperate to achieve goals in a simple planning environment. The planning environment was a 'world' consisting of four objects, whose positions define the 'world state': John (a robot), Mary (a robot), a door and a bolt. The robots can be IN or OUT depending on which side of the door they are. Within this world, there are three simple actions which change its state given certain conditions, and there are a number of possible goals which John and Mary possess. These actions, their consequences and preconditions can in fact be summarised as simple 'STRIPS' (Fikes & Nilsson 1971) planning operators, such as:

action: MOVE

preconditions: door open

effects: robot moves

action: PUSH

preconditions: bolt UP

effects: door changes position (OPEN<-->CLOSED)

action: SLIDE

preconditions: robot IN

effects: bolt changes position (UP<-->DOWN)

The model for each robot contains specific versions of these operators, in terms of beliefs concerning preconditions and effects of actions, together with a set of goals. Robots are agreed on the high-level goal, but have different capabilities and beliefs on effects of actions, which has important consequences on the dialogue generated to facilitate cooperative achievement of an agreed goal. Given these representations, a simple planner is applied to generate a tree of subgoals to be achieved *for each robot* and the actions required to achieve them. Dialogue is therefore generated as part of plan execution in those circumstances where cooperation is required. For example, in order for John to achieve his goal of JOHN IN he requires the subgoal DOOR OPEN to be achieved, which requires sliding of the bolt, which in turn can only be done by a robot who is IN. Since John is OUT and Mary IN, John requires Mary's cooperation. What is therefore required is that the robots are agreed on *goals, plans to achieve goals, and individual responsibility for performing actions* to achieve goals. In Power's system, this cooperation in planning is facilitated by the use of *conversational procedures* attached to the fundamental planning procedures:

"The robots are capable of planning separately as well as cooperatively; they only co-operate when one of them has a goal which he fails to achieve on his own ... during co-operative planning some goals on the planning tree are labelled as joint responsibilities ... The usual result is that they employ conversational procedures at these points in order to perform subtasks which would otherwise have been performed by planning procedures." (Power 1979, p. 121)

Conversational procedures are basically *adjacency pairs* (Schegloff & Sacks 1973), where for "... each type of adjacency pair that they use, the robots have a list of instructions which lays down how each utterance in the pair should be produced and interpreted." (Power 1979, p. 112). The following conversational procedures are incorporated into the model:

AGREEGOAL - secure cooperations of partner in achieving goal

AGREEPLAN - agree plan to achieve goal which is joint responsibility

ASK - obtain information

TELL- give information

DISCUSS - compare beliefs about consequences of actions

ASSESS - assess result of action performed

There is one further conversational procedure which has important implications:

ANNOUNCE - explicitly announce the call of a conversational procedure.

Power adopts the method of *explicit announcement* (i.e. a form of explicit goal negotiation) of conversational procedures, since for the conversation to proceed effectively, participants must have compatible representations of the dialogue. In human dialogue this is normally done by a participant simply beginning to use a procedure, making utterances which can be interpreted as implicit negotiation (in the case of AGREEGOAL or AGREEPLAN) or announcement of the procedure. He remarks that

"... this method raises complicated problems of inference which are beyond the scope of the model. These problems are therefore evaded by the method of announcing conversational procedures explicitly..."

(pp. 126-127)

Management of updating participants' models and controlling the alternation of speakers and listener roles in executing conversational procedures is performed in the program by a "chairman", for which Power makes no theoretical claims.

In the second section of his 1979 paper, Power states clearly how the program advances our understanding of dialogue, together with criticisms and proposals for an improved model. The main respect in which he claims that the program is an advance on other language-using programs, is that

"... it is able to represent the point of an utterance." (p. 131).

This leads to the question of "... How well does it represent the point of an utterance?" (p. 131). The proposals for an improved model involve detailed discussion of the notion of 'Speech Acts', as derived from the philosophy of language of Austin (1962), and the later development by Searle (1969). It is beyond the scope of our review to enter into detailed analysis of this area of philosophy of language. Power's major (self) criticism is that although the robots have a representation of which utterances to use in order to achieve cooperative goals, they have no explicit representation of knowledge concerning *how* these specific utterances perform the (speech) acts which they do, and so *how* they achieve the goals which they do. The importance of establishing a connection between goals and linguistic utterances designed to achieve them is clear if the program is to be used as a model of *human-computer* dialogue, as Power claims it can (p. 148). The solution strays deep into the heart of speech act theory; but we may say that it fundamentally depends on constraining the set of possible goals that a speaker could be referring to, to those which are not already achieved and could be construed as part of a candidate plan, and establishing further 'inference rules' to interrelate possible goals via concepts of 'belief', 'intention', 'willingness' and so on.

The importance of Power's work lies in its integration of planning physical actions and the resultant generation of utterances which attempt to achieve cooperation on achievement of joint cooperative goals. It does not, however, address effectively the problem of recognition of goals

and plans in language (how can rules be provided for every possible utterance which could map onto intentions?), the problem of planning sequences of *utterances* or *linguistic goals*, and the related problem of non-explicit recognition of mutual goals.

In Power's theory, conversational procedures are not themselves combined into linguistic plans, but are 'side-effects' of plans concerning physical actions in some world. The work of Cohen and Perrault (1979) considered this problem of combining intentional dialogue units represented as 'speech acts' into plans. Beginning from the initial formulation of speech acts by Searle (1969), they attempted to define planning operators for 'requesting', 'questioning' and 'informing', and to specify a set of 'metatheoretical' principles for how such definitions should be formulated. Operators are formulated in a form similar to STRIPS (Fikes & Nilsson 1971), and the 'frame problem' of specifying which aspects of the world are *not* affected by operators is similarly avoided¹⁵. The following is their first attempt at defining the REQUEST operator:

"

REQUEST(SPEAKER, HEARER, ACT)

CANDO.PR: speaker believe hearer cando act

and

speaker believe

hearer believe hearer cando act

WANT.PR speaker believe speaker want (request-instance)

EFFECT: hearer believe

speaker believe speaker want act

"

(Cohen & Perrault 1979, p. 190)

There are a number of things to note about this definition. Firstly, it embodies the author's "Point of View Principle" for construction of speech act operators, which states simply that

"... Since the applicability conditions affect the planning of that speech act, the preconditions are stated as conditions on the speaker's beliefs and goals. Correspondingly, the effects describe changes to the hearer's mental state."

(Cohen & Perrault 1979, p. 189)

For this reason, preconditions always begin with "speaker believe", and effects always begin with "hearer believe". The second thing to note is that the effects of a speaker requesting that a hearer do some act are *not* that the hearer then wants to do the act, but rather that the hearer believes that the speaker wants him to do the act. The same problem applies to INFORMing: the hearer does not therefore believe the belief informed, but simply believes that the speaker believes it. In order to 'bridge this gap', they insert a "mediating step ... that

¹⁵ The so called 'STRIPS assumption' that all aspects of the world remain unchanged except those described by an operator's effects and their logical entailments.

models what it takes to get someone to want to do something" (p. 190). The new operator "... trivialises the process it is intended to model by proposing that to get someone to want to do something, one need only get that person to know that you want them to do it." (p. 190). Given this initial definition of REQUEST, and subsequent definitions of INFORM, a number of modifications to preconditions of REQUEST have to be made in order to ensure consistency, since a REQUEST should be equivalent to an INFORM of a *want*. The principal modification lies in removing the CANDO precondition 'speaker believe hearer believe hearer cando act' from request, and inserting it as a precondition of the 'bridging' operator CAUSE-TO-WANT. A further symmetrical relationship between INFORM and REQUEST which they develop is the definition of a QUESTION as a REQUEST to INFORM some information. Given these final definitions, the operators can be used to develop plans of speech acts to perform for goals whereby speakers want to alter the belief states others.

The work of Cohen and Perrault demonstrates that *if we could define speech acts as planning operators, then they can be combined into plans "... whose effects are primarily on the models that speakers and hearers maintain of each other."* (p. 179). They do not address the problems of the "triviality" (their term) of many of the operators - what is it for someone to be 'convinced'? - nor of recognition of possible speech acts in natural language, and generation of natural language from speech acts. These two problems were addressed in the subsequent work of Allen & Perrault (1980), on "Analyzing Intention in Utterances", and in the work of Appelt (1982) on generating natural language utterances from high-level speech-act formulations.

3.4.5.2 Inferring Goals and Plans in Dialogue

Allen and Perrault (1980) considered the problem of inferring a speaker's plan of speech acts - as formulated in the earlier work of Cohen and Perrault (1979) - from *questions* asked in the restricted domain of a passenger making enquiries at a railway station booth. The specific problem considered is to give an account of

"... what we think occurs when one agent A asks a question of another agent B which B then answers."

(Allen & Perrault 1980, p. 146)

The situation considered bears some resemblance to Power's (1979) notion that discourse is generated when one agent cannot achieve one of the subgoals of his plan without cooperative action from the other speaker. For Allen and Perrault these are considered as "obstacles" in A's plan, the identification of which will lead the respondent B to plan utterances in an attempt to achieve that inferred subgoal:

"A has a *goal* to acquire some information; he creates a plan (*plan construction*) that involves asking B a question whose answer will provide the information A then executes his

plan, asking B the question. B receives the question, and attempts to infer A's plan (*plan inference*). In this plan, there may be goals that A cannot achieve without assistance. These are the *obstacles* in A's plan. B can accept some of these obstacles as his own goals and create a plan to achieve them. B's response is generated when he executes this plan".

(Allen & Perrault 1980, p. 146)

A problem arises, however, when we consider the fact that not all obstacles are *explicit* in the sense of being subgoals of an inferred plan: if someone asks "Where is the nearest garage?" and the respondent replies "on the next corner", if the respondent knows that the garage is closed, then the provision of this *implicit* information would be a normal part of human cooperative behaviour.

The process of plan inference starts from the assumption that from the questioner's utterance a single *observed action* (i.e. speech act operator) or an *expected goal* has been identified. From the predefined set of operators (for INFORM, REQUEST, and so on), this enables the inference of an initial *incomplete plan*. Possible additions to the incomplete plan generate a set of *alternative partial plans* by the operation of a set of 'plausible inference rules'. For example, "if A wants *n* to want to do some action ACT, then A may want *n* to do ACT" (P. 154), and "... if A want to know whether a proposition P is true, then it is possible that A want to achieve a goal that requires P to be true..." (p. 155). A set of *heuristics* are then used to select between the alternative partial plans, by assigning a 'rating' to each. For example, they use a heuristic which prefers plans which are 'simpler' than others, in terms of having more preconditions currently satisfied:

" (H1) Decrease the rating of a partial plan if it contains an action whose preconditions are false at the time the action starts executing."
(p. 157).

How, then, are goals and speech acts identified from an utterance in the first place? The method relies on looking for "surface syntactic clues" (p. 171), which may themselves generate a set of *alternative possible speech acts*, and in turn increase the number of partial plans to be considered:

"... there will often be cases where a mood ambiguity cannot be resolved at the syntactic level, and in these cases the alternative will be enumerated and each case will become a plan alternative. Since the number of surface speech acts is small, this approach is reasonable."
(Allen & Perrault, 1980, p. 171)

As with most artificial intelligence programs, power is achieved at the expense of domain restriction. In the approach of Allen and Perrault, it is assumed that the set of possible goals is restricted in the domain. In more open domains we have no way of knowing whether the heuristics described would be able to constrain a combinatorial explosion when the number of

possible speech acts becomes less constrained:

"Our specification of the actual plan inference process, however, is not detailed enough to allow it to perform in more complex domains than the train station. Considerable work needs to be done to specify more control heuristics. Large domains probably require the introduction of domain-specific inference rules"

(op cit p. 176)

As with Sidner's (1985) work on intention recognition, they look to models of dialogue *focus* to help constrain the range of possible expectations:

"One of the major problems in larger domains is the effective management of the large number of potential expectations. The progress of a dialogue relies on old expectations, but more importantly, establishes new ones. Grosz (1977) shows how task structure can limit these expectations. Topic shifts and the initial dialogue expectations remain unaccounted for."

(Allen & Perrault 1980, p. 176)

In conclusion, Allen and Perrault identified the relationship between speech act analysis at the syntactic level and the underlying planning system, as one of the most difficult problems facing their theory (Allen & Perrault 1980, p. 176, para. 7). More recent research which can be viewed as a continuation of the planning and speech act approach has concentrated on the more fundamental problems of the logical analysis of the concept of *intention*, and its relationships with beliefs, plans, goals and actions. In the most general sense, 'dialogue' in the way in which it is commonly understood as a linguistic communicative and cooperative act, is viewed as just one consequence of a more general theory of *rational agency* (Kiss 1986). Cohen and Levesque (1987) have thus aimed to develop a carefully worked out theory of *rational action*, which provides

"... both a logic in which to write specifications for autonomous agents, and an initial theory cast in that logic."

(Cohen & Levesque 1987, p.229)

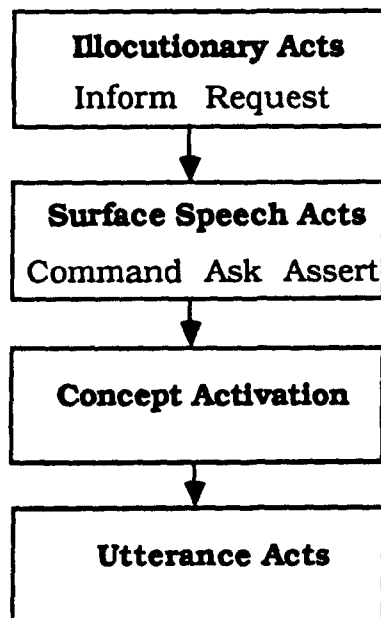
They argue that in existing planning systems which are concerned with the planning of communication acts needed for one agent to affect the mental state and behaviour of another, a theory of intention is only implicit in the agent's plans, goals and the architecture of the planning system. They claim that this makes it difficult to reason formally with intentions, and the fact that there is no fixed specification of intentions which is implemented in the planner has the effect that if the architecture changes, then so may the implicit definition of intentionality. We shall not discuss the details of their logic of intention here, since our concerns are to discuss general trends in dialogue research *per se* which are relevant to intelligent tutoring systems. The logic used is based on a possible worlds formalism to represent the state of affairs intended, and includes modal operators for beliefs, goals and the

representation of events which are imminent and just occurred. Important features of their analysis include the idea of "rational balance" in intentions, in the sense that agents should rationally *persist* with certain intentions over time, yet not drop them precipitously, that agents should rationally only intend to achieve something which they believe to be possible, and that agents need not intend all the expected side-effects of their intentions. Their work can thus be seen as the beginnings of a formal theory of the relationship between belief, intentions and goals, which must be an important theoretical underpinning for any theory of dialogue.

3.4.5.3 Planning Utterances to Achieve Multiple Goals

The work of Appelt (1982; 1985) also included a sophisticated logic for reasoning about beliefs in planning utterances. He concentrated on specifying the detailed relationship between high level descriptions of a speaker's goals and generation of low-level sentences. In particular, he concentrated on the problem of "Planning Natural-Language Utterances to Satisfy Multiple Goals" (PhD thesis title, 1982). His 'KAMP' system plans referring expressions in English given an initial specification of a speaker's goals in terms of high level 'illocutionary acts', a representation of the participants' knowledge, and the knowledge contained in an expert system for assembling air compressors (the domain considered is thus similar to Grosz's task oriented dialogues). Using a hierarchical planner (similar to that described by Sacerdoti 1977), it successively expands the goal into a set of subgoals, to finally generate an English sentence which incorporates all the information and speech acts required to satisfy all subgoals. It expands by moving down the hierarchy of linguistic actions shown in figure 3.3:

Figure 3.3
Hierarchy of linguistic actions, described by Appelt (1983).



The system is best described by considering an example. Consider the example where KAMP starts from the goal 'remove the pump from the platform', as represented in terms of a logic of knowledge and action:

True (\sim Attached(Pu,PL)).

The planner creates a *procedural network* containing this single plan step, which can be thought of as a two dimensional network in which the horizontal represents temporal ordering of actions, and the vertical dimension represents successive abstraction. In order to plan speech acts to affect the conversant's knowledge, the system requires and provides a powerful logical reasoning mechanism, based on a 'possible world' semantics for a meta-language which refers to an 'object language' of intensional operators such as 'know' and 'intend'. The possible worlds fulfill a similar function to representations of the 'world state' in 'conventional' planning systems, and the operators which transform one state into another are thus represented as object-language relations which map one state onto another. As each planning node is expanded, a new 'possible world' state is assigned to it, the corresponding relations between it and the previous world are asserted, and powerful deduction systems can then attempt to prove that the goal would be satisfied in the newly assigned possible world. For any goal that succeeds (or is already true), it is marked as a "phantom goal", but is kept in the plan since later planned actions may make it no longer hold - in which case corrective action can be taken to reachieve or preserve it. Like Sacerdoti's (1977) system, KAMP uses a set of 'critics' to examine the plan, and check for interactions between actions which would require the plan to be modified. In our example, the goal

True (\sim Attached(Pu,PL)).

might be expanded to the (unordered) subgoals

IntendsToDo(John,Remove(PU,PL))

Do(John,Remove(PU,PL))

Do(Rob,Request(John, Remove(PU,PL)))

since the system has decided that Rob should request that John perform the action of removing the pump because a request is the only possible action one agent can perform to affect another's intentions. So far, the system will have planned a set of actions at the level of 'illocutionary acts', and proved that they will achieve their goal. Next the planner needs to plan *surface speech acts*, which may involve reasoning about when a speech act will be recognised as a particular illocutionary act. For example, the 'request' illocutionary act in the subgoal

Do(Rob,Request(John, Remove(PU,PL)))

needs to be recognised as a request for an *action* rather than for information. In this case a

speech act corresponding to *utterance of an imperative sentence* is planned. Again, 'plan critics' are employed at each hierarchical level of expansion in order to check for action interference. Now the plan is expanded to the level of "concept activation". This entails essentially planning which *concepts* need to be *referred to* in each surface speech act. This is not simply a matter of uttering all referents in the speech act, since some may be already referred to in some other part of the plan, or already assumed by the participants - for example, the system could say "remove it from the platform" rather than "remove the *pump* from it". In our example, suppose that the reasoning component decides that the intended proposition

.Remove(PU,PL)

requires that both the concepts 'platform' and 'wheelpuller' need to be "activated" or mentioned. The final level is to choose a basic syntactic structure which fits the propositional content and the surface speech act - i.e. an imperative sentence with the structure 'V NP (PP)' and propositional content @ (PU) & @ (PL), where the '@' symbol indicates referring expressions. Critics now examine the whole hierarchical plan, an example being the "action-subsumption" critic, which essentially checks that actions are not planned for effects which are already achieved:

"An action A_1 *subsumes* an action A_2 if A_1 and A_2 are part of the same plan, and action A_1 (in addition to producing the effects for which it was planned), also produces the effects for which A_2 was intended. Therefore, the resulting plan need only include action A_1 (and its expansion) to achieve all the goals."

(Appelt 1985, p.22)

Thus the action-subsumption critic examines pairs of adjacent illocutionary acts - such as requesting and informing - and checks whether one could be subsumed by another. Action subsumption is important in *allowing a single sentence to achieve multiple goals*. Suppose the following illocutionary acts are present in an initial plan:

request 'Remove the pump'

inform 'The wrench is in the toolbox'

Subsumption critics could notice that at the linguistic level, the verb 'to remove' can have an 'instrument' case, which specifies the instrument which does the removing. The **inform** illocutionary act can now be subsumed by the **request** illocutionary act, and subsequent planning levels adjusted to generate the single utterance

"Remove the pump with the wrench in the tool-box".

We have discussed Appelt's work in some detail, because this is essential to understand its very important contribution to the main issue which has been discussed in this section: *how are topic or focus and intention in dialogue interrelated ?* Appelt contrasts his approach to language generation with the view adopted by previous systems, which he terms the "conduit metaphor" (Appelt 1982, p. 7):

"The conduit metaphor refers to the treatment of language as a pipeline or conduit that transfers information between the speaker and the hearer. The speaker has some idea of what he wants to say ... , he encodes that idea in natural language ... , sends the package through the conduit to the hearer, who unwraps the package and removes the contents.

The disadvantage of this general view is that it forces one to acknowledge a very strong separation between the two stages of the language-planning process: deciding what to say and how to say it "

(Appelt 1982, pp. 7-8)

Appelt therefore views a speaker's *goals* in language as in some sense primary, with decisions as to *what* to talk about as distributed throughout a hierarchical set of planning decisions. This is the case because reasoning about goals requires reasoning about mutual knowledge, belief, desires and intentions of conversants. His model concerns the planning of a *single utterance* by a *single conversant* , and so does not directly address issues of how interactive dialogue is managed in terms of cooperative goal pursuit, recognition of intention, and recognition and planning of dialogue focus. We may recall, however, the 'conjecture' of Grosz and Sidner (1986) that what we mean by the 'focus' of a dialogue, is closely related to its goal rather than its subject or topic. His work does seem to have established that in planning dialogue, decisions about focus and intention can not be considered in isolation, since they interact and mutually constrain each other in complex ways, the full understanding of which remains an issue for future research. As we remarked earlier, there does seem to be a growing consensus amongst researchers that changes in intentional structure are the primary determinants of dialogue structure, that focus shifts are produced as side-effects, and that the range of possible focus shifts in turn provides a constraint on possible intentional shifts (Grosz & Sidner 1986). Such a general view of the primacy of intentional structures and the distribution of decisions concerning what to say, is compatible with the dialogue model described in chapter 5 of this thesis. In that model, the primary decision made is the 'dialogue move' to perform, but issues of topics, affect that decision in terms of topics discussed previously, the belief states of dialogue participants, and the relative focussing on topics.

Finally, in this section on 'goal directed' approaches to language, we may ask whether Appelt's approach to planning a single utterance by a single speaker could be applied to planning of *dialogue* , by the simple extension of providing a (still) higher level planner for a sequence of illocutionary acts? The question would ultimately require a thorough review of the extensive literature on AI planning systems, which is beyond the scope of this chapter; but we shall restrict ourselves to an identification of the major issues involved.

Appelt's system addresses the planning problem of achieving multiple simultaneous goals of a *single speaker* by essentially using the 'plan critics' approach advocated by Sacerdoti (1977). In an interactive dialogue, a single speaker would need to satisfy the multiple simultaneous goals of *both speakers* in order to secure cooperativity in performance of dialogue actions. This involves numerous unsolved research problems, not least concerning those of inferring intention from a speaker's utterances - i.e. knowing what their goals are in order to plan for their achievement - and reasoning/modelling what their beliefs are in order to plan for them. A second problem concerns the so called 'frame problem', and the STRIPS assumption that actions have all and only their presumed effects. If we have no reasonable theories of belief revision, then in the context of planning linguistic acts to changes a hearer's beliefs, the STRIPS assumption is not adequate. Let us suppose that such a plan could in fact be constructed: how is the production of a complete or partially complete plan for the utterances of a single participant to be managed within the context of an *interactive* dialogue, where each participant has the freedom to interrupt and pursue dialogue goals which may change the (mental) 'world state' of participants? It is conceivable that plans could be interrupted, and a decision made to replan or 'repair' the original plan. In a truly interactive dialogue, where both partners have equal freedom to pursue and negotiate their goals, might not a complete plan for a section of dialogue turn out to be so much wasted computational effort? The work of Peachey and McCalla (1986) is an example of such a planning approach from the point of view of Intelligent Tutoring Systems. It uses a set of STRIPS-like (Fikes & Nilsson 1973) operators for *concept* teaching goals, which produce a 'curriculum plan' of topics to be taught. From the point of view of dialogue research, there are two simple objections to this approach:

- (1) it depends on a separation of planning what to teach, from the problem of how it should be taught, without specifying the interrelationship between the two;
- (2) the very notion of a complete plan for a complete interaction presumes that the student can play no part in influencing the course of that plan.

The work of Grosz and Sidner (1986) specified a relationship between changes in intentional structure and in focus structure, but left open the question as to how these intentional units are interrelated. The influential work of Suchman (1985) provides an alternative view to the notion of 'planning' which has the possibility of being applied to intention in dialogue, and more specifically in intelligent teaching dialogues. Her thesis contrasts the predominant view in Cognitive Science that recognition and production of plans as the prerequisites for physical and linguistic action, with her view - derived from social science - of plans as *situated action*. She argues against the notion of plans as being formed prior to action, 'in the head' of an agent, which can therefore be recognised and used to guide action, in favour of plans as essentially "*ad hoc*", and "derivative from actions *in situ*" (Suchman 1985, p. 35). The conventional view of planning is that it involves nearly all the computational effort, and that execution is largely mechanical and in a sense 'trivial'. When plans are considered as situated action, this very distinction between planning and execution is eroded - plans become the post hoc rationalisations of actions once they are under way, and are largely considered only in cases where the effects of actions become in some way incomprehensible. In applying this view to

dialogue generation, an essentially reactive and *opportunistic* view of decision making results: we decide on what to say in the situated context of the dialogue once it is under way. We seem to constrain our choices of linguistic actions in terms of their perceived contribution in a given context to some set of very general high level objectives, in a manner similar to the 'Trukese navigator' described by Suchman:

"The European navigator begins with a plan - a course - which he has charted according to that plan. His effort throughout his voyage is directed to remaining 'on course.' If unexpected events occur, he must first alter the plan, then respond accordingly. The Trukese navigator begins with an objective rather than a plan. He sets off toward the objective and responds to conditions as they arise in an *ad hoc* fashion."

(Suchman 1985, p. 1)

Exactly *how* we constrain our actions according to the significance that we assign to a particular context "... is the outstanding problem." (Suchman 1985, p. 46). Elsom-Cook (1989a) has suggested that Suchman has adopted a somewhat simplistic characterisation of 'AI planning' against which to propose her alternative theory. He argues that dialogue 'planning' in ITS may be modelled by 'opportunistic planning' (Hayes-Roth et al 1979), which pursues multiple partial plans, and which can incorporate 'bottom-up' local interaction needs into 'top-down' partially formed plans.

3.4.6 Computational Models of Dialogue : Summary

Let us now summarise the issues which computational approaches to modelling dialogue have addressed. The central issue for all approaches concerns

the identification of a set of supra-sentential dialogue segments and the specification of a set of relations between them.

Grammatical approaches identify segments analogous to sentence-level syntactic categories, based on units of the *domain*, of dialogue function, or of both. In interactive dialogues, the notion of grammatical well-formedness seems to be contradicted by the occurrence of intelligible ill-formed sequences. A number of approaches conflate domain based units with functional units into a single representation which provides inadequate theoretical tools for modelling dialogue. Most recent approaches identify segments based on *two* features:

- (1) 'focus'/the domain/topic,
- (2) goals/function/intentions.

An exception is the work of Reichman (1985), who attempts to combine the two into a single segment type - the "context space" - with a grammatical relationship between segments. The further problems are then generated of specifying the relations between segments for both categories, and of specifying how the two theoretically separable structures interrelate. In the field of dialogue focus, the dominant theory is that of Grosz (1977), who described focus in terms of partitioning of a semantic network, with focus shifts constrained by the 'task' discussed. In

the generalisation of her theory to incorporate intentional structures (Grosz & Sidner 1986), intentions were included in focus spaces, which thus provided local *constraints* on shifts in separable intentional structure. Their theory says nothing about how intentional shifts take place. The view that intentional structure, and its changes, is primary is shared by most authors (Grosz & Sidner 1986; Levin & Moore 1977), along with the view that the two issues of focus and intention can not each be considered in isolation. The dominant approach to specifying the relationship between intentional units has been based on *planning* a succession of *speech acts*. Research in this area has considered the problems of representing speech acts as planning operators (Cohen & Perrault 1979), the inference of plans from recognised goals and speech acts (Allen & Perrault 1980; Sidner 1985), models of intention and rational agents (Cohen & Levesque 1987; Kiss 1986) and the planning of natural language utterances from high-level illocutionary acts (Appelt 1982). Problems facing these approaches include representing, modelling and reasoning with conversants' knowledge and beliefs, and constraining the range of possible speech acts which may be inferred (given even that intentional units can be initially segmented), and the resultant space of possible plans which may be inferred. Allen and Perrault (1980) initially address these problems by a severe domain restriction and use of plan inference heuristics. Appelt (1982;1985) argues that no simple separation can be made between planning what to say and how to say it, since hierarchical levels of planning mutually constrain each other in both directions, and we may need to reason about knowledge and the way it is expressed at any number of these levels. In interactive dialogue, the necessity for mutual recognition and cooperative agreement on pursuit of goals arises. This raises the problem of implementing a mechanism for *implicit negotiation* of goals (Levin & Moore 1977; Power 1979), which relates to the problem of intention recognition. There are alternative approaches to viewing the relationship between intentional units other than as components of plans. In extending Appelt's approach beyond the utterance level to interactive dialogues, problems of plan execution monitoring and allowing mutual freedom to cooperate and pursue dialogue goals may speak in favour of an opportunistic approach which is closer to Suchman's (1985) notion of *situated action*.

3.5 Conclusion

Research into intelligent teaching dialogues needs to address the following fundamental questions and findings from dialogue research:

- (1) Grammatical and schematic approaches are inapplicable to interactive dialogue since they conflate fundamental theoretical distinctions between topical and intentional structures, and posit an inapplicable relationship between dialogue segments.
- (2) Focus and intention require theoretically separable analyses.
- (3) The relationship between focus and intention is complex and must be addressed by any dialogue model.
- (4) Dialogue 'topic' seems to be more closely identified with goal structures than with subject matter.

- (5) Focussing in dialogue may include focussing on intention.
- (6) Intentional shifts seem to be primary, and take place in the context of local focus constraints.
- (7) In cooperative dialogues, mutual goals or intentions are primarily communicated *implicitly*. The modelling of implicit negotiation relates to intention recognition.
- (8) Intention and plan recognition in terms of speech acts faces the problem of constraining a very large space of possible intentions and plans if it is to move beyond severe domain restrictions.
- (9) Planning in interactive dialogues faces the problems of securing cooperativity by negotiation, execution monitoring, replanning, and reasoning about participants' knowledge and beliefs. Situated action provides an alternative theory for the connection between intentional dialogue units.

These are some of the issues which face future research into modelling intelligent teaching dialogues. In the next chapter we shall describe a model of intelligent teaching dialogues called 'Negotiated Tutoring'. The approach provides a first step in the direction of applying theories of dialogue to generate intelligent teaching interactions which give primacy to intentional units (called 'dialogue moves') within focus-based constraints, and attempt to secure greater symmetry and student freedom in the interaction by providing a common mechanism for negotiating student and system interaction goals. Chapter 5 describes an implementation of some of the aims of the Negotiated Tutoring approach, which generates dialogues called 'critical arguments'. Given the point of view described in this chapter on discourse genres, it will be evident that 'critical arguments' are dialogues in a very general sense, and are individuated from other kinds of dialogue by the range of intentional units which they provide, rather than by idiosyncratic generation mechanisms.

Chapter 4: Negotiated Tutoring - an approach to intelligent tutorial interaction

"Good learning situations and successful ITS are successful not because they enable a learner to ingest preformed knowledge in some optimal way, but because they provide initially underdetermined threadbare concepts to which, through conversation, negotiation and authentic activity, a learner adds texture."

(John Seely-Brown, "Toward a New Epistemology of Learning", p.4, 1989)

4.0 Introduction

This chapter is concerned with describing an approach to the generation of tutorial dialogues, called 'Negotiated Tutoring'. Although relevant literature will be mentioned in passing, it will not be reviewed in detail: we shall concentrate on describing the analytical tools which underlie the approach and some aspects of its implementation in detail.

Any approach to human-computer dialogue must be integrated with a model of the changing course of the participants' knowledge and beliefs throughout an interaction. The approach must include a participant's knowledge and beliefs concerning their own and the other's beliefs in the 'domain of discourse', their goals, intentions, and conceptions of connectedness or coherence in dialogue. We have implemented some aspects of our approach to tutorial dialogue within an architecture of an Intelligent Tutoring System ('ITS') called 'KANT' (Kritical Argument Negotiated Tutoring system), in order to test its consistency, completeness and feasibility. KANT is a model which generates *critical argument human-computer dialogues, with an educational purpose*, within domains the whole or part of which can be characterised as justified belief. The domain chosen to illustrate the interaction model is the set of justified beliefs concerning the presence of musical structures in a given melody, described in chapters 2 and 5 of this thesis. The approach is specified in general terms, so that the system could engage in an educational interaction in any other set of justified beliefs which may be similarly represented. We shall therefore discuss a hypothetical dialogue in a simple area of geology to illustrate this feature. In the future description of the model, we shall use the term 'KANT' to refer to the *abstract model* for tutorial dialogues, which is not identical with the degree to which it is instantiated in an implementation. The abstract model is defined within this chapter.

4.1 Models of high-level dialogue

Before describing KANT in the next chapter, we need to be clear about the nature and purpose of models of human-computer dialogue, and their relationship with implementation in computer programs. KANT is intended as a model of what Kiss (1986) terms 'high-level dialogue': it is a

model of the structure of interaction between rational agents who are able to cooperatively pursue their respective goals in the interaction, taking into account the intentions and knowledge of the other participant. In this sense, a 'dialogue action' designed to achieve a conversant's goal is not necessarily equivalent to a generated linguistic utterance: in a computational environment, the system's goal of explaining some concept to the user, for example, may be achieved by generating text, graphical examples, or examples in other media appropriate to the domain in question (in our case, this could include generation of musical examples via a synthesiser linked to the computer system). KANT is therefore concerned with modelling the high-level structure of interactions between two participants, termed the 'initiator' and the 'respondent' (whose roles may both be taken by either a computer-based teaching system or an interacting human student), rather than being concerned with natural language generation down to the level of the sentence. When high-level dialogue structures are expressed linguistically, they generally span several sentences. The structures generated by KANT include intentional or goal-based units and associated metadialogue structures to negotiate cooperative pursuit of these goals, with an independent structure of 'topic' in terms of concepts explicitly mentioned in the dialogue.

4.2 Primacy of the interaction model

We use the term 'KANT' to refer to both a dialogue model and an Intelligent Tutoring System. This statement expresses a specific point of view regarding the primacy of the model of the interaction in Intelligent Tutoring. Most ITS's have been essentially centred around an existing knowledge representation for a domain, with interaction mechanisms designed to *express* that domain in some way, giving a dialogue structure essentially determined by traversal of the domain and its communication to the student with a set of limited forms of interaction. For example, GUIDON (Clancey 1982, 1987) was built around an existing set of production rules, and SCHOLAR (Carbonell 1970, Collins, Warnock & Passafiume 1974) used the state-of-the-art knowledge representation of its time - Quillian's (1968) theory of semantic networks - to essentially question the student on topics whose succession was determined by searching the network (with attached numerical 'importance' tags). Elsom-Cook has recently argued (1989a) that intelligent tutoring should focus on the needs of the interaction, including student and domain models, since the interaction is our primary source of observable information which we have concerning the teaching process¹, and there is little point in developing sophisticated student modelling techniques, for example, which play little role in the actual interaction. In KANT, therefore, the modelling technique used for the domain (frame-based 'plausible reasoning') was chosen so that it was based on psychological theories² of a human

¹ We are not making the behaviourist claim that the observable interaction is the only phenomena which can be said to exist in a scientific sense, but rather stating that since human-human and human-computer tutorial dialogues are observable, they form a reasonable focus for research - especially given the relative neglect of this area in current ITS research (as described in chapter 3 of this thesis).

² It was based on versions of schema theory (Bartlett 1932) applied to music cognition (McAdams 1987; Stoffer 1985), as described in chapter 2 of this thesis. In addition, the model incorporated

perceiver's unconscious inference of structural boundaries in music, and a dialogue participant's conscious recall of this knowledge as a set of uncertain justified beliefs. This is precisely the kind of belief concerning which a human dialogue participant would engage in the kind of dialogue which we have termed 'critical argument'. The domain representation is thus designed to serve the needs of the interaction, since its representation as declarative knowledge which was assumed certain could have led to a quite different interaction style, inappropriate in this domain. From a psychological perspective, the domain and student knowledge representations must serve the needs of the interaction, given an obvious relationship between a human being's production and understanding of dialogue and their access of relevant knowledge in memory: if the dialogue structure is to be reasonably coherent and comprehensible, then it must embody a close relationship with a cognitive model of the domain. In order to be 'comprehensible', a human-computer dialogue model does not have to attempt to be *identical* with human-human dialogues. Given the present restricted human-computer 'communication channel' in commonly available information technology (usually typed input), there are of necessity a number of ways in which human-computer dialogues will be quite different from human-human dialogues. For example, dialogues in human settings possess a microstructure of minute pauses in real-time, concerning succession of speaker turns (Levinson 1983), which can not presently be simulated on essentially non-realtime execution of computer programs on machines with limited memory. Also, in human-computer dialogues we can not presently model effectively the role of non-verbal communication and adoption of social roles in a given cultural context. Our aim has been to model as many features of human dialogues as possible, within the limits of existing technology (a kind of dialogue which Petrie-Brown (1989) has termed "Intelligent Teaching Dialogue").

4.3 Research methodology

KANT was developed as an attempt to synthesise a number of research directions in Intelligent Tutoring Systems and in computational modelling of dialogue, rather than from analysis of human teaching dialogues (such as in the work of Collins and Stevens, 1983). There are a number of reasons for adopting this research methodology. Apart from our distinction between human teaching dialogues and intelligent tutoring dialogues, the main reason for producing computational models from the existing literature on Intelligent Tutoring Systems and on dialogue from the linguistics literature (Levinson 1983), is that *we presently lack sufficiently precise theoretical models for the analysis of dialogue protocols*. The iterative refinement of such models seems to be presently the best way of producing theories, which could subsequently be used to analyse protocols. Few suitable theories exist at present: Levinson (1983), for example, cites only the model of Power (1979) as an example of a new approach in discourse analysis. Other models include those of Elsom-Cook (1984) and Reichmann (1985); but these models do not provide us with a model of a cooperative interaction between two autonomous agents. KANT aims to take features of these and other partial models, and extend them to incorporate other features which present analysts in the linguistics literature have identified as important features in dialogue (for example, in our inclusion of user symmetry of interaction goals, and cooperative negotiation). We have thus drawn extensively on the literature from

"plausible reasoning" mechanisms based on those described for medical diagnosis (Pauker et al, 1984).

computational modelling of dialogue in the development of KANT, guided by the belief that another feature of the centrality of the interaction model in intelligent tutoring is the fact that teaching dialogues are *dialogues* in a general sense (apart from having educational goals). By implementing the model in a computer program we are able to express theories more precisely and completely. We have not, however, extended implementation of the model to the development of a fully robust tutoring system which could actually be used and evaluated with students, nor has the model yet been tested on dialogue protocols (see chapter 6, §6.3 of this thesis, on further work). Self has recently (1989) called into question the view implicit in much ITS research, that ITS architectures should be fully implemented and evaluated in real educational contexts: if our goal is to develop a theoretical model, then much refinement can be achieved by making predictions about the model (*re* theory), and executing the model to test these predictions. Predictions will be confirmed to the extent that the model exhibits the appropriate desired new behaviours, which may be initially tested according to the extent that they match our linguistic intuitions concerning conversational coherency (a common approach in linguistics).

4.4 Design goals: Negotiated Tutoring

Negotiated Tutoring aims to generate tutorial dialogues in which the student is treated as far as possible as an equal participant in a cooperative interaction. This includes symmetrical representations of student and system beliefs and conceptions of dialogue focus, and the freedom for each to negotiate pursuit of teaching and learning goals via a set of 'dialogue moves' (goals), sequences of which form alternative interaction styles. The approach synthesises a number of current trends in ITS research, some of which have been identified by Self (1988):

- (i) provision of a wider range of interaction styles;
- (ii) increased symmetry in the interaction by extending the range of possible interaction types available to the student;
- (iii) an emphasis on cognitive skills such as reason and argument, and on metacognitive skills such as reflection upon and possible revision of beliefs, rather than 'education as knowledge communication'.

These design goals can be viewed as corresponding to some elements of what Bennett (1976) terms a "progressive" teaching style, which emphasises features such as

"Teacher as guide to educational experiences ... active pupil role ... Pupils participate in curriculum planning"
(Bennett 1976, p. 38)

In addition, the Negotiated Tutoring approach has affinities with the recent 'social cognition' approach (Seely-Brown 1989) to learning environments, in terms of its emphasis on negotiation in dialogue as a mechanism for the cooperative joint *construction* of knowledge.

We shall discuss each of these design goals of Negotiated Tutoring in turn.

(i) *multiple interaction styles*

In KANT interaction styles are represented as alternative sequences within a hierarchical tree of dialogue moves, as will be discussed in the next chapter. We need to be clear about the nature and purpose of such interaction or teaching styles, and the way in which teaching styles relate to dialogue in general. When we look at the ITS literature we find the terms 'strategy', 'style' and 'approach' used almost interchangeably. One of the few pieces of research which has addressed the problem of clarifying these terms is the work on the DOMINIE system (Elsom-Cook & Spensley 1988). DOMINIE embodies (at least) eight *styles*, of which examples are "cognitive apprenticeship", "successive refinement" and "discovery learning", together with a general *strategy* consisting of a set of rules, which are " ... intended to embody a strategy of choosing the least interventionist method which is applicable" (Elsom-Cook 1989a, p.10).

Cognitive apprenticeship consists essentially of bottom-up traversal of the domain representation, where each domain unit is taught by "watching the expert in action and asking questions", then asking the student to repeat the expert's actions. Similarly, successive refinement traverses the domain top-down using a simple presentation mechanism for knowledge units and sub-units. It seems therefore, that in simple terms, for many cases,

$$\begin{aligned} \text{teaching style ('ts')} = \\ \text{domain traversal procedure}^3 \text{ ('dt')} + \\ \text{domain interaction style ('di')} \end{aligned}$$

For example, we can view the earlier work of Collins and Stevens on the 'Socratic teaching' style as

$$\begin{aligned} \text{dt} &= \text{top-down / topic importance heuristic} \\ \text{di} &= \text{question / answer} \end{aligned}$$

What is meant (in the existing literature) by an 'approach' therefore, seems to be a general (philosophical?) preference for a *single* interaction mechanism (eg "elicitation", "questioning"), the choice of which thereby selects the domain traversal procedure with which it is conventionally paired. We may therefore ask whether viewing the problem as that of defining strategies in terms of choice among existing traversal/interaction style pairings will be a flexible enough approach to tutorial interaction if we want to specify a relationship between work on teaching styles and on dialogue *per se*. If we look at the literature on computational models of dialogue, we find a variety of super-sentential dialogue units which have been identified - 'conversational moves' (Reichman 1985), 'dialogue games' (Levin & Moore 1977), 'speech acts' (Cohen & Perrault 1979), and so on. How can we find a principled way of choosing amongst alternative analyses of this kind? The problem is important since the expression of a teaching style must at some level consist of a succession of dialogue units. What exactly are the special characteristics of educational discourse, as well as those characteristics

³ Elsom-Cook & Spensley (1988) term this "search and validation".

shared with units analysed in other more general kinds of dialogue? In KANT there are the relatively crude beginnings of an attempt to represent teaching styles as embedded implicitly in sequences of dialogue units. The fundamental domain traversal algorithm remains *the same*⁴ for all sequences of dialogue moves (i.e. we do not have a fixed set of traversal/dialogue move pairings), and consists of local focus constraints defined by a spreading activation model. The situation is therefore more complex, and could be approximately represented as follows:

Negotiated Tutoring :

Negotiated Tutoring approach =

domain traversal algorithm (dt):

'choose the next most locally relevant concept, according to a spreading activation theory of dialogue focus'

+

'negotiate acceptance of the goal of discussing that concept')

+

interaction type(s) (di):

'choose a set of possible dialogue goal to pursue, which are satisfiable in the current dialogue state'

+

'choose a dialogue goal based on a set of educational principles'

+

'negotiate acceptance of the dialogue goal')

The general idea is that the dialogue may follow its own internal dynamic, constrained by focus-based coherence. As focus shifts, the interaction style problem is considered afresh, so that we have a flexible combination of dialogue moves, which may or may not be those combinations which correspond to teaching styles used by human teachers. This is a relatively crude analysis, since in a sense the method of analysis of teaching approaches into a domain traversal algorithm and an interaction type is not really applicable to Negotiated Tutoring as an approach: each of these elements is controlled by a set of factors, and negotiation is a primary mechanism involved in the selection procedure for each. The consideration of each dialogue participant as an autonomous agent means, therefore, that domain traversal is not simply controlled by the decisions of *the system*. At the end of the previous chapter we noted that approaches to computational modelling of dialogue had converged on the view that the problems of deciding 'what' to say and 'how' to say it were closely linked, and that there is no simple distinction to be made between the two levels of decisions. In the Negotiated Tutoring approach, therefore, we can see that the processes of choosing concept teaching goals and the interaction types which achieve this are *interrelated* : a decision as to an interaction type (such as a 'challenge') may select a particular concept to refer to (such as one previously mentioned), and dialogue focus shifts may suggest a concept to be discussed, where the interaction type used depends on those used previously. We have stated the approach in terms

⁴ In further work (chapter 6) we propose the inclusion of a set of domain traversal algorithms.

of existing conceptions of teaching styles and strategies for purposes of clarity and comparison. The detailed implementation of the approach will be described in the next chapter.

The necessity of providing suitably defined multiple interaction styles follows from the simple fact that the student and teacher may have alternate preferred ways of learning and teaching. The general point to be made is that we need to be clear about what teaching styles are, what they could be, and how they relate to dialogue in general. Such a process of clarification is of course one of the benefits of AI models of the teaching process, and should ultimately leave us free to consider the problems of practical effectiveness of teaching styles and the acceptability of attendant educational philosophies.

The provision of multiple interaction styles generates the problem for an ITS of choosing between them at any point in an interaction. Any approach to answering this question depends on our definition of teaching or interaction styles in the first place. If we take Elsom-Cook's conception of 'strategies' then these must derive from 'educational principles', such as

'prefer to elicit rather than instruct'

'prefer the least interventionist applicable strategy'

If we could show that non-interventionist strategies resulted in students remembering less 'facts' this would only be a problem if the goal of such strategies was to inculcate more of the beliefs of the teaching system (a view which, as Self (1989) argues, is contrary to most of the major educational philosophies of this century). If we pose the question of strategy choice in this way then it seems that all we can ask of researchers who are forced to develop and formalise their own set of educational principles is that they are sufficiently clear and explicit, in terms of goals, supporting philosophical arguments for acceptability of those goals, and the methods defined to achieve those goals. The necessity of representing these principles explicitly is not only a question of research methodology. In a recent extensive analysis of classroom teaching dialogues, Edwards & Mercer (1987) have argued that correction of simple factual errors is usually a relatively small and trivial component of such dialogues. Important issues concern the learner's understanding of what they term the "ground rules of educational discourse", of which examples are

"1 it is the teacher who asks the questions

2 the teacher knows the answers

3 repeated questions imply wrong answers"

(Edwards & Mercer 1987, p. 45)

From an analysis of extensive classroom dialogue protocols, Edwards and Mercer claim that a number of major misconceptions in educational discourse a lack of such a set of shared assumptions, which are always kept implicit. They therefore argue that there may be a case for making these rules explicit. In intelligent computer-based teaching dialogues, students may come to the interaction with a *different* set of assumptions, but even in this case

misunderstandings may arise. In these cases, making the abilities, limits of understanding and expectations in the interaction explicit and clear may well avoid a number of mutual misunderstandings.

Finally, we may ask how important is our question concerning the definition of *teaching strategies*? If the teacher controls the interaction completely, and enforces learning goals as teaching goals, then the question is extremely important. If the student is equally free to negotiate learning and interaction goals, then the problem reduces to a problem faced by the *tutoring system* - a single participant in the dialogue.

In summary of this discussion of the role of multiple interaction styles in ITS, we have made the following points:

- 1 The terms 'teaching style', 'teaching approach' and 'strategy' as they are currently used in the ITS literature require clarification and definition.
- 2 We have proposed that a current definition of *teaching styles* seems to be that they are pairings of an algorithm for traversing a domain representation, together with a single *interaction type*.
- 3 Teaching *strategies* seem to be general educational principles which select preferred *teaching styles*. Strategies are difficult to define and relate to general philosophies of education. Given this definition of *teaching styles*, strategies therefore select a single domain traversal algorithm and its conventionally paired interaction types.
- 4 We argue that teaching styles must at some level reduce to a set of interactional units at the *dialogue* level, together with domain traversal mechanisms, and that we require a flexible combination of multiple interaction types with multiple domain traversal procedures. These combinations may not necessarily correspond to existing conceptions of teaching styles in ITS.
- 5 The Negotiated Tutoring approach is not easily analysable in this way, since the processes of choosing concept teaching goals and the interaction types which achieve this are *interrelated*, principally by a process of *negotiation* of concept and dialogue goals. Educational principles operate to constrain the possible satisfiable dialogue goals which could be applicable in a given dialogue state. The domain is traversed in terms of local dialogue coherence, defined by a spreading activation model of semantic memory, given negotiation and acceptance.
- 6 Mutual misunderstanding in tutorial dialogues may result from a lack of explicit shared understanding of roles and goals in the interaction. There may therefore be a case for making these assumptions explicit.

(ii) *interaction symmetry*

Interaction symmetry is important if we wish to move away from the approach of viewing the student's learning goals as the simple inverse of the system's teaching goals⁵, and a view of

⁵ In our review of existing research in computational generation of dialogue, we described how existing research makes this assumption *implicitly* in simply not providing the opportunity for the student to genuinely influence the course of the interaction. We are not claiming that the view of the student's goals as the inverse of those of the system is made *explicitly* by these researchers, but rather that the limitations of present research in fact imply this view.

the student as a largely passive respondent to the system's teaching plans, towards a more symmetrical model of interaction between dialogue participants. In KANT we adopted the initial simplification of presenting the same set of dialogue moves to the student, and allowing the student to similarly negotiate cooperative pursuit of these goals. In conclusion to this chapter we shall argue that *symmetry* presupposes *cooperativity* (Grice 1975) in dialogue, to resolve possible conflict in the interaction goals symmetrically provided to each dialogue participant, which they may attempt to pursue by *negotiation*. The student is, however, constrained to express beliefs about the domain in terms of the *system's* representation, which raises a question which faces any model of computer-based interaction in terms of understanding unrestricted input, and the student's 'viewpoint' on the domain (Moyse 1989). A question is raised here (discussed as further research) which concerns the possibility of managing the interaction of possibly conflicting goals and plans on the part of each participant, in a situation where each is given sufficient freedom to negotiate, to cooperate, or not to cooperate: how can symmetrical freedom to negotiate and cooperate be integrated with the pursuit of coherent plans and goals on the part of either dialogue participant?

(iii) *cognitive and metacognitive skill acquisition*

KANT aims to teach the cognitive skill of arguing for one's justified beliefs in a 'reasonable' manner. It is worth emphasising this apparently minor departure from the view of ITS as concerned with "knowledge communication" (Wenger 1987): KANT teaches a skill in use of *dialogue* ('critical argument') by participation in that *dialogue itself*. We further assume that in the course of such a dialogue either participant may reflect upon the dialogue so far, and possibly revise their beliefs (a *metacognitive* activity). We therefore require psychologically plausible models of when an argument is more 'convincing' than another, and when and how a dialogue participant should revise their beliefs⁶. There is a difference between reasoning as a cognitive process and the cognitive skill of expressing unconscious reasoning in the form of a reasoned argument in dialogue (see §3.3 of the previous chapter). The explicit teaching of some formal logical or mathematical inference procedure is therefore quite different from the explicit teaching of how to present a reasoned argument in dialogue. Even formal reasoning procedures are usually taught by visual demonstration, accompanied by a teaching dialogue. We therefore need to develop clear models of this relationship between reason and argument (*qua* language and thought) if we are to use the interaction model as the primary vehicle for teaching such cognitive and metacognitive skills.

4.5 Negotiation, symmetry and cognitive skills in dialogue

Let us now attempt to introduce these design goals of Negotiated Tutoring by considering a *hypothetical* dialogue. Our aim is to introduce the features of Negotiated Tutoring discussed so far in terms of an analysis of this dialogue. We shall consider a dialogue in a simple area of *geology*, rather than in the domain of analysis of musical structures⁷, to enable a consideration features of our model which does not demand knowledge of musical analysis on

⁶ Our research has not, of course, provided any such psychologically valid models for argumentation. Such an enterprise would be a major research project in Cognitive Science.

⁷ Examples of dialogues in the domain of musical structures are given in chapter 5 of this thesis.

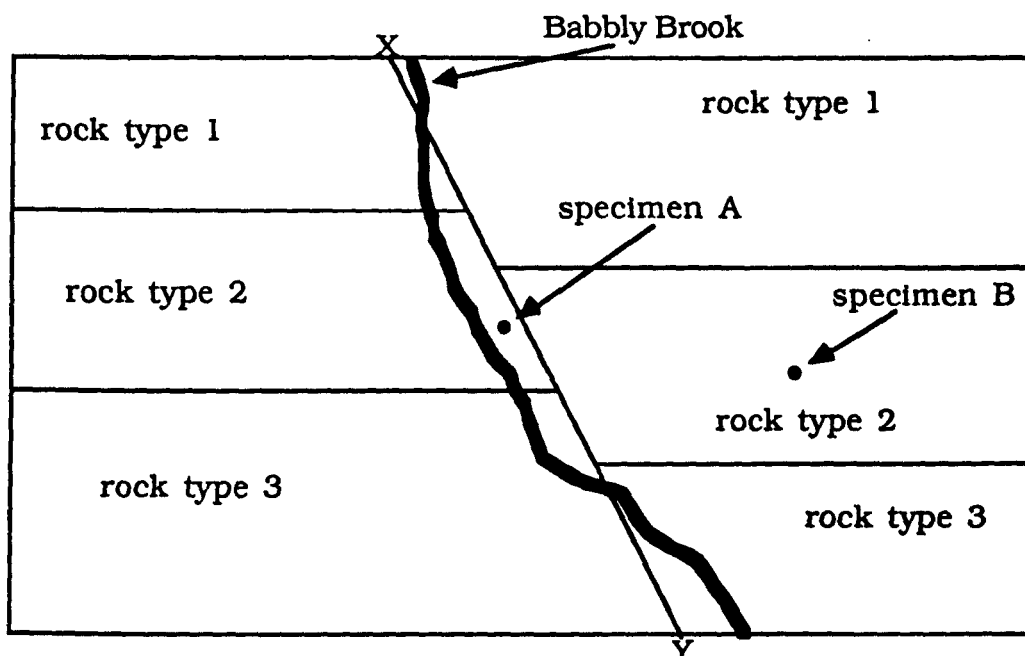
the part of the reader. We must emphasise that the Negotiated Tutoring approach was not *based on* an analysis of such hypothetical dialogues. Our approach is based on a synthesis of a number of features of existing research in computational generation of dialogue, which we describe in chapters 3 and 5 of this thesis. In view of these facts, *we clearly can not claim psychological plausibility for our model*.⁸

Figure 4.1 is a simplified geological map, showing positions where rock specimens have been collected. The following (fabricated) dialogue is concerned with identifying the specimens in a discussion between two dialogue participants ('A' and 'B'). The general tutorial scenario is that A and B are looking at the map and the rock specimens which have been collected at the positions marked on it. They engage in a discussion about the nature and origin of the specimens.

⁸ We reiterate that the dialogue is an *invented* one for the purposes of describing a number of the features which we have attempted to integrate into our model, and that we can *not* therefore claim any *psychological plausibility* for our model of dialogue. The model which is partially implemented in KANT was not as such *derived* from such invented examples: it was based on a synthesis of existing research in ITS, computational linguistics, explanation and argument research, which we described in detail in the previous chapter. *The invented dialogue is therefore used here merely as an informal way of introducing a number of features of the Negotiated Tutoring model.* There are a number of methodological reasons why we did not consider it appropriate to collect and analyse dialogue protocols at this stage in the development of our research. An important feature of our research was the claim that negotiation phases should be introduced into tutorial dialogues generated by Intelligent Tutoring Systems. At the present state of research, the only work which has made the beginnings of achieving this is the relatively early work of Power (1979), who had to resort to the use of *explicit announcement* for negotiating joint goals. Since, as Levin and Moore (1977) argued, most negotiation is *implicit*, we would not expect to be able to identify this in written dialogue transcripts, and it would furthermore be *excluded* by computer-mediated dialogues. Our research therefore concentrated on establishing a role for negotiation phases within a general ITS architecture. The problem of modelling implicit negotiation is left for further research, as discussed in the final chapter of this thesis. We argue that it is a very difficult problem which faces human-computer interaction research in general, and which would require extensive research in psychosocial analysis of non-verbal communication, which was beyond the scope of our present research domain. Finally, we argued earlier in this chapter that since we presently lack formal models with which to analyse dialogue protocols, our research is a necessary first step towards their development.

Figure 4.1

A simplified geological map, showing positions where rock specimens were collected



Fabricated dialogue about simple geology

1 A: Why don't we talk about specimen B?

2 B: Ok

3 A: Have you any idea what it might be?

4 B: Yes; I reckon it's limestone?

5 A: I agree, but why?

6 B: I'm not sure really - it just looks that way. What is limestone exactly, anyway?

7 A: Well, it's a sedimentary rock composed mainly of calcium carbonate. Calcium carbonate is derived from...

8 B: What's 'sedimentary' mean?

9 A: It means that the rock is produced by the deposition of 'sediment' which is subsequently turned into a rock by a process known as 'diagenesis'. I could go on and tell you about how sediments become sedimentary rocks in diagenesis, but maybe you can now tell me why you think it's limestone?

10 B: Not really; can you tell me what the sediments of limestone are?

11 A: Ok; but I'd really rather you gave me a more reasonable answer to why you think it's limestone. Think about what I said earlier about limestone: it's obvious!

12 B: Oh yeah; sorry; they must be sediments of calcium carbonate. Oh; I get it now; that would be why it fizzed when I put some HCL on it. I must be limestone because the calcium carbonate evolves carbon dioxide when you put acid on it.

13 A: Hold on a minute: specimen A fizzes as well: are you saying that it's limestone as well?

14 B: Well they can't both be, I suppose, they look completely different.

15 A: Let me tell you what I think about specimen B, OK?

16 B: Go on then.

17 A: If you look at it under the hand-lens you can actually see the individual sediment grains. You can also see a few fragments of a fossil, which I think is ...

18 B: Yeah, I've seen that one before. I guess its shell must be made of calcium carbonate, right?

19 A: Yes. Now you should be able to tell me what specimen A is..

20 B: I'll have a go. I reckon it must be quartz because I can see the crystals.

21 A: Well, I don't think so. Quartz is made of silicon dioxide, which wouldn't fizz. The cubic crystal forms suggest to me that it's in fact calcite. There are also other reasons. Look at the position of the bedding: why do you think the stream flows where it does?

22 B: That's the easiest, softest bit for it to flow in.

23 A: The beds are obviously displaced by a fault along X-Y, and such movements generate the heat necessary to recrystallize the limestone into another form of calcium carbonate. What type of rock do you think type 1 is therefore?

24 B: I have no idea...

25 A: Think about it. Let's go back and think about sedimentary rocks and diagenesis, ok?

26 B: Alright; you said that limestone was sedimentary?? What were you going to say about some fossil?

27 A: I was going to say that it's called *Micraster* ...

28 B: Yeah I know; *Micraster* is a quite rare echinoderm isn't it?

29 A: I think so...

30 B: You can tell it's that because it's heart shaped so that it could dredge through the silt at the bottom easily, sifting out bits of food. It's reasonably rare I think.

31 A: If you know that *Micraster* occurred in shallow seas, and you think about what can happen in them, what sediments might type 1 be composed of.

32 B: Sand?

33 A: Could be. So the rock could be sandstone.

<<continues>>

There are a number of features worthy of note which occur in our invented dialogue:

- (1) Both participants are freely able to negotiate the pursuit of their goals in the interaction ;
- (2) either participant is able to interrupt and take control of the interaction;
- (3) both participants use a range of interaction types ;
- (4) before attempting to pursue a dialogue goal, each participant negotiates cooperation from the other ;
- (5) participant A obviously seems in a position of greater knowledge ;
- (6) when stating justifications for belief, a participant may try to go on and explain their justification further ;
- (7) both participants are able to tell the other facts they did not know;
- (8) A is not simply concerned with telling B all that she knows, but wants B to discover things for him/herself;
- (9) when participants have successfully 'gained the floor' in dialogue, they sometimes simply want the other to speak.

Let us perform a simple analysis of large scale structures in the dialogue, as a way of explaining a number of features of Negotiated Tutoring. First we introduce some terms which are part of the model of tutorial dialogue partially implemented in KANT:

n = the current negotiator, the current participant who begins to negotiate the next goal in dialogue

s = the person who pursues the goal as speaker

c = the current topic

i = the current instance of c

$g(n, g, s, c)$ = negotiator n , speaker s , pursues goal g about topic c

$n(n, g, s, c)$ = n negotiates goal g concept c , speaker s

Fabricated dialogue about simple geology: Analysis

1 A: Why don't we talk about specimen B?

A decides that B can understand a discussion about concept 'limestone', and can integrate this with existing knowledge; negotiates this.

$n(A, \sim, \text{specimen_B})$

2 B: Ok

B cooperates with proposed negotiation

3 A: Have you any idea what it might be?

$n(A, \text{concrete_claim}, B, \text{specimen_B})$

("A negotiates that B makes a concrete claim about specimen_B and B is pursuing the goal")

4 B: Yes; I reckon it's limestone?

g(A, concrete_claim, B, specimen_B)

("A has successfully negotiated that B make a concrete claim about specimen B")

5 A: I agree, but why?

g(A, agree_claim, A, specimen_B)

n(A, support_concrete_claim, B, limestone)

6 B: I'm not sure really - it just looks that way. What is limestone exactly, anyway?

B can not cooperate; negotiation fails; B negotiates new goal

n(B, abstract_claim, A, limestone)

7 A: Well, it's a sedimentary rock composed mainly of calcium carbonate. Calcium carbonate is derived from...

A responds to negotiation with acceptance

g(B, abstract_claim, A, limestone)

Now A decides that B will not understand this simple explanation without the discussion of some required precursory knowledge; A attempts to continue pursuit of this goal by pursuing a further goal:

g(B, g(B, abstract_claim, A, calcium_carbonate), A, limestone)

8 B: What's 'sedimentary' mean?

B interrupts A's dialogue action - cooperativity was not achieved, since B wanted to know about the term 'sedimentary' instead. B's response to A's presumption of cooperativity is to respond with a specific form of non-acceptance - the negotiation of a new goal:

n(B, abstract_claim, A, sedimentary)

9 A: It means that the rock is produced by the deposition of 'sediment' which is subsequently turned into a rock by a process known as 'diagenesis'. I could go on and tell you about how sediments become sedimentary rocks in diagenesis, but maybe you can now tell me why you think it's limestone?

A responds with acceptance, performs action to achieve dialogue goal:

g(B, abstract_claim, A, sedimentary)

Considers possibility of negotiating a new goal to discuss required subconcepts, but decides to attempt to renegotiate previous failed goal

n(A, support_concrete_claim, B, limestone)

10 B: Not really; can you tell me what the sediments of limestone are?

B can not cooperate; negotiates a new goal:

n(B, abstract_claim, A, sediment)

11 A: Ok; but I'd really rather you gave me a more reasonable answer to why you think it's limestone. Think about what I said earlier about limestone: it's obvious!

Non-acceptance response, renegotiates

n(A, support_concrete_claim, B, limestone)

12 B: Oh yeah; sorry; they must be sediments of calcium carbonate. Oh; I

get it now; that would be why it fizzed when I put some HCL on it. I must be limestone because the calcium carbonate evolves carbon dioxide when you put acid on it.

B responds with acceptance;

g(A, support_concrete_claim, B, acid_reaction)

13 A: Hold on a minute: specimen A fizzes as well: are you saying that it's limestone as well?

A responds quickly, without negotiation

g(A, complementary_claim, A, specimen_A)

14 B: Well they can't both be, I suppose, they look completely different.

B considers revising beliefs

15 A: Let me tell you what I think about specimen B, OK?

n(A, concrete_claim, A, specimen_B)

16 B: <<raises eyebrows>>

A interprets this as non-verbal acceptance

17 A: If you look at it under the hand-lens you can actually see the individual sediment grains. You can also see a few fragments of a fossil, which I think is ...

g(A, support_concrete_claim, A, specimen_B)

18 B: Yeah, I've seen that one before. I guess its shell must be made of calcium carbonate, right?

g(B, support_concrete_claim, B, specimen_B)

19 A: Yes. Now you should be able to tell me what specimen A is..

n(A, concrete_claim, B, specimen_B)

20 B: I'll have a go. I reckon it must be quartz because I can see the crystals.

g(B, concrete_claim, B, specimen_B)

21 A: Well, I don't think so.

g(A, disagree_claim, A, specimen_B)

Quartz is made of silicon dioxide, which wouldn't fizz.

g(A, concrete_claim, A, quartz)

The cubic crystal forms suggest to me that it's in fact calcite.

g(A, concrete_claim, A, calcite)

g(A, support_concrete_claim, A, calcite)

There are also other reasons. Look at the position of the bedding: why do you think the stream flows where it does?

n(A, support_concrete_claim, B, specimen_B)

22 B: That's the easiest, softest bit for it to flow in.

Acceptance

g(A, support_concrete_claim, B, calcite)

23 A: The beds are obviously displaced by a fault along X-Y, and such movements generate the heat necessary to recrystallize the limestone

into another form of calcium carbonate.

`g(A, abstract_claim, A, calcite)`

What type of rock do you think type 1 is therefore?

`n(A, concrete_claim, B, type_1)`

24 B: I have no idea...

`non-acceptance`

25 A: Think about it. Let's go back and think about sedimentary rocks and diagenesis, ok?

`n(A, discuss, sedimentary_rock)`

26 B: Alright; you said that limestone was sedimentary?? What were you going to say about some fossil?

`acceptance`

`n(B, claim, A, fossil)`

27 A: I was going to say that it's called *Micraster* ...

`acceptance`

`g(B, concrete_claim, A, fossil)`

28 B: Yeah I know; *Micraster* is a quite rare echinoderm isn't it?

`g(B, agree_claim, B, Micraster)`

`g(B, abstract_claim, B, Micraster)`

29 A: I think so...

`g(A, agree_claim, A, Micraster)`

30 B: You can tell it's that because it's heart shaped so that it could dredge through the silt at the bottom easily, sifting out bits of food. It's reasonably rare I think.

`g(B, support_abstract_claim, B, Micraster)`

31 A: If you know that *Micraster* occurred in shallow seas, and you think about what can happen in them, what sediments might type 1 be composed of.

`n(A, concrete_claim, B, type_1)`

32 B: Sand?

`acceptance`

`g(B, concrete_claim, B, type_1)`

33 A: Could be. So the rock could be sandstone.

`g(A, agree_claim, A, type_1)`

<<continues>>

The above analysis shows a very rich structure in an apparently simple dialogue. These are not the only structures present, nor is this the only possible method of analysis: we have produced a partial analysis of the dialogue in terms of the theoretical constructs of KANT, and will discuss affinities with related methods of analysis in the next chapter. At present we are simply concerned with describing the method of analysis to be used in the later description of

KANT. The dialogue shows both conversants pursuing interaction goals in the dialogue using a range of symmetrical interaction styles, negotiating pursuit of these goals using explicit and non-verbal communication, negotiating goals at different levels of abstraction and considering revising their beliefs in the course of the dialogue. The number of interaction goals required by our analysis is very few: they generate a variety of interaction styles by their use and negotiation by each speaker. There are many other features present in the dialogue - both participants are clearly doing some *reasoning*, which they only partially express in dialogue (lines 22 & 23, for example); but we shall not deal with these features at present. Let us expand our discussion of these features identified in the analysis before describing how dialogues displaying some of these features may be generated.

4.6 Negotiated Tutoring features in the example dialogue

(1) Symmetrical pursuit of interaction goals in the dialogue

Both A and B are able to interact to pursue their goals in the dialogue, even though A seems to be the 'teacher': B is not simply a passive respondent. For example, in lines 1-5, A pursues the general goal of eliciting some response from B, but in line 6 B decides to pursue the goal of asking A about one of the concepts ("limestone") which A mentioned.

(2) Range of interaction styles

Both A and B possess a range of interaction styles which are available to both participants. These interaction styles are analysed as a set of simple goal types with different speaker roles. The simple set used in the dialogue are as follows:

ELICITATION:

`g(x, concrete_claim, y, inst(c))`

`g(x, concrete_claim, y, supports(inst(c)))`

Negotiator x wants speaker y (not equal to x) to make some claim about a specific instance of concept c, or about justifications for an instance of c.

[lines 3, 11]

QUESTIONING:

`g(x, claim, y, c)`

Negotiator x wants speaker y (not equal to x) to make some general or abstract claim about the nature of concept c.

[line 6]

INFORMING:

`g(x, claim, x, inst(c))`

Negotiator x wants x to state a belief about a specific instance of concept c.

[lines 4,20,21]

SUPPORTING:

`g(x, claim, x, supports(inst(c)))`

Negotiator x wants x to state instances of supports for an instance of c.

[lines 12,17,18,21]

EXPLAINING:

`g(x, abstract_claim, x, c)`

Negotiator x wants x to make a general claim about concept c.

[lines 7,21,23,28]

JUSTIFYING:

g(x, claim, x, supports(c))

Negotiator x wants x to make a general claim about the kinds of supporting evidence relevant to instances of concept c.

[line 9]

CHALLENGING:

g(x, disagree_claim, x, inst(c))

Negotiator x wants x to pursue the goal of disagreeing with some concrete claim about concept c (the claim could have been stated by either participant)

[line 21]

g(x, agree_claim, x, inst(c))

Negotiator x wants x to pursue the goal of agreeing with some concrete claim about concept c.

[lines 28,29,33]

g(x, complementary_claim, x, supports(inst(c)))

Negotiator x wants x to make a claim complementary to supports stated for an instance claim (made by either participant). A complementary claim is one concerning some other concept instance (where the new concept may or may not be equal to the one previously stated) which possesses the stated supports as a subset of its supports.

[line 13]

g(x, new_supports, x, supports(inst(c)))

Negotiator x wants x to state some supports for instance of c, not previously stated.

[line 17]

g(x, disagree_supports, x, supports(inst(c)))

Negotiator x wants x to disagree with a set of supports for a previously stated instance of c.

[line 21]

g(x, agree_supports, x, supports(inst(c)))

Negotiator x wants x to agree with a set of supports for a previously stated instance of c.

[line 19]

Clearly, we have not exhausted all the possibilities of speaker and negotiator substitution here, nor are these the only possible kinds of goal which may be pursued in dialogue - these are simply the goals used to analyse this dialogue fragment. The number of goal-types is very small, giving considerable theoretical parsimony with a larger range of substitution possibilities and interaction types.

A further point to note is that although we have termed this dialogue a 'critical argument', it

is clear that given a range of interaction styles and the ability of both participants to freely negotiate, that any interaction type could follow any other: critical argument, in our sense, should not be understood as a form of "discourse genre" (Reichman 1985), such as 'explanation' or 'Socratic dialogue' - it is dialogue *sui generis*. The Negotiated Tutoring approach is therefore not identical with the specific set of dialogue goals identified for characterising 'critical arguments': other goals could be substituted whilst retaining the essential features of the approach. We simply claim that some such set of dialogue-level goals can characterise aspects of teaching dialogues.

(3) Negotiation of goal pursuit

In this dialogue, a set of utterances can be identified which are not concerned with belief communication, but which are concerned with negotiating the cooperation of the other participant in pursuit of a proposed dialogue goal to be jointly pursued. These utterances form sequences of metadialogue, concerning the future roles and form of the dialogue itself, rather than with belief communication by dialogue goal pursuit. There seem to be three possibilities here:

(i) *Participants presume cooperativity* in pursuit of their goals in the absence of interruption (line 7). This reveals the crucial role of timing in dialogue, where at 'transition relevance points' (Schegloff & Sacks 1973), a speaker presumes to 'hold the floor' unless interrupted.

(ii) *Participants negotiate explicitly* (line 9), in which case there can be two responses: acceptance or non-acceptance. In the case of acceptance, this can be explicit (line 2), or non-verbal (line 16). Non-acceptance is either signalled by the respondent explicitly signalling this (line 6), or else by their beginning to pursue or negotiate some new goal (line 8). These phenomena of expectation of acceptance, and non-acceptance by performing another goal have been described by Levinson (1983) in a discussion of the functional role of adjacency pairs (Schegloff & Sacks 1973). The adjacency pairs occurring here have the function of negotiating future dialogue utterances.

(iii) *Participants make a tentative utterance* which communicates their intentions, giving the other the possibility of interrupting (lines 1 & 27). These utterances comprise a special set of metadialogue adjacency pairs called *preannouncements*. They are designed to preannounce the topic to come, generated when the speaker is uncertain of the respondent's knowledge, giving the respondent opportunity to preempt if the preannounced utterance will not tell them something new. This phenomenon relates to Grice's (1975) 'maxim of quantity' - "make your contribution as informative as is required".

This phase of negotiation must always be present in dialogues which conform to Grice's (op. cit.) maxims of cooperativity: it is either more or less explicit, non-verbal, or simply presumed, thus having no external expression.

(4) Goal abstraction, specificity, and recursion

We have described those utterances in the dialogue which are not concerned with negotiation as *dialogue goals*. By this, we are presuming that each participant is a rational agent who possesses goals in entering into interaction, and who make utterances to achieve them: goals concern communication of beliefs between participants, and dialogue actions are utterances

which express those beliefs. In negotiation, however, goals are expressed with greater or lesser degree of abstraction. For example, a participant may negotiate that some "claim" be made, without saying exactly what sort of claim. These matters may be refined further as the dialogue progresses. Similarly, goals may be expressed incompletely - for example, a participant may say "let's talk about x", without specifying who will say what and how. In general, it seems reasonable that goals, may be successively refined in specificity, from general to less general, and that in the course of this refinement, possible future speaker and concept roles may be clarified. We will speak of the negotiator, speaker, concept and instance attached to a dialogue goal as the *parameters* of the goal (Levin & Moore 1977; Power 1979).

A further point is worth mentioning about the succession of dialogue goals: we mentioned that *any goal may follow any other*, contingent on cooperativity being achieved. This approach must be viewed in contrast to grammatical (Robinson 1982; Reichman 1985) and schematic (Bobrow et al, 1977; Stutt 1987) approaches to dialogue, described in the previous chapter. However, in line 7 we had a case where the execution a dialogue action to achieve a goal, resulted in the recursive pursuit of a further goal:

`g(A, g(A, abstract_claim, A, calcium_carbonate) , A, limestone)`

In this example, A pursues the goal of making an abstract explanation of the concept 'calcium_carbonate' in order to make an abstract explanation about limestone. In other words, part of the performance of one dialogue action was the performance of another. This recursive descent clearly has limits in terms of the knowledge of the participants, and the ability of participants to perceive this as coherent. We term such a pairing of the negotiation phase for some goal, which the performance of the dialogue action designed to achieve the goal a *negotiated turn*.

(5) Belief revision, cognitive skills and the 'domain of discourse'

Before two conversants can argue critically in a domain, they clearly have to have some understanding of the domain of discourse. There are a number of utterances in the dialogue above which are therefore concerned with belief communication (lines 7,9, etc.). At other points in the dialogue it is clear that this is not the only function of utterances: at certain points conversant A is concerned with trying to make B reflect on her beliefs and possibly revise them (line 13). In addition, by engaging in the dialogue, we might say that both conversants are practising the cognitive skill of generating dialogue in a certain ordered or 'reasonable' form (they are required to justify their beliefs, for example, rather than simply have them accepted as they stand - line 11). We can therefore isolate three functions or activities occurring in dialogues of this kind:

- (i) communication of the domain of discourse (understanding possible evidence sources);
- (ii) the metacognitive skill of a participant reflecting upon and possibly revising beliefs;
- (iii) the cognitive skill of conversing in a certain way within mutual expectations.

Again, we would emphasise that dialogues of this kind do not necessarily conform to some ideal

'schema' or set of 'well-formedness rules': they reflect the conversants' *expectations* concerning 'reasonable' discourse, which may or may not be fulfilled.

4.7 Knowledge and belief in dialogue

Finally, we may ask what kinds of knowledge and beliefs would be required on the part of each dialogue participant in order to engage in such dialogues? In the dialogue above we can hypothesise at the very least that participants require:

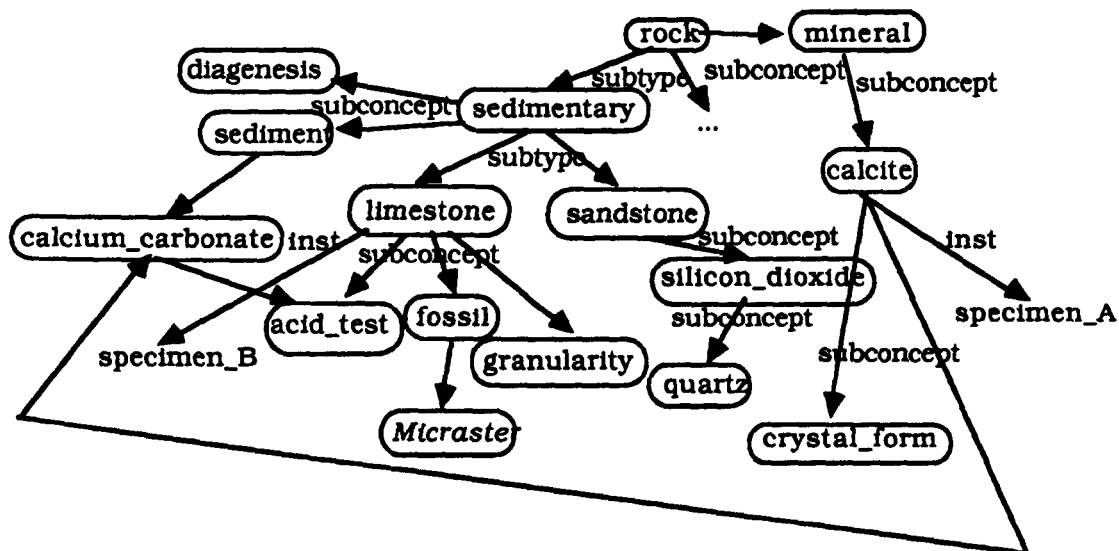
- (i) knowledge and beliefs concerning the domain of discourse, including possible sources of justifying evidence, deeper understanding of the concepts involved, and knowledge of how these concepts interrelate;
- (ii) a set of reasoning methods by which they use this knowledge to decide which set of beliefs they currently subscribe to;
- (iii) some memory of what has already been said in the dialogue;
- (iv) some knowledge of possible ways of expressing their goals in the dialogue, and of what may be relevantly said at any point;
- (v) beliefs concerning the beliefs of the other participant.

All of these kinds of knowledge and belief will be variously portioned between what is currently being stored or rehearsed in working memory, and what is stored in long-term memory. The factual knowledge of each participant can be represented in the following fragments of semantic networks (Quillan 1968) - figure 4.2 shows the beliefs of participant A, and we assume that the knowledge and beliefs of participant B may be similarly represented⁹. For example, the belief that specimen B is limestone is represented by a node 'specimen B', linked to the node 'limestone', by an 'instance' link type. Questions concerning the definition of 'knowledge' and 'belief' are *philosophical* questions, and as such form part of ongoing research. In terms of our representation, 'knowledge' and 'belief' are represented as *traces* in semantic memory, where a specific belief that some instance is an instance of a particular concept is simply represented as an 'instance' slot in the node for that concept. The status of such a representation as 'belief' or as ('certain') knowledge (or 'true belief') is therefore determined by the *function* which it serves in *dialogue*. We have argued that appropriate dialogue structures for memory representations which are considered as 'beliefs' correspond to dialogue moves for *critical argument*, within a Negotiated Tutoring approach to educational interactions.

⁹ This is, of course, a considerable simplification, which we make given the focus of our research on dialogue generation mechanisms. Future research must consider the possibility that a conversant may have a different conceptualisation of the domain. Important consequences may follow for the kind of negotiation in which a conversant engages, since in explanation dialogues, for example, conversants may engage in cooperative negotiation of *meaning* and the kind of explanation which is mutually required (O'Malley 1987b).

Figure 4.2

Part of a semantic network for knowledge and beliefs of hypothetical dialogue participant A



4.8 Summary of design goals in Negotiated Tutoring

We have analysed structures occurring in a *fabricated* dialogue in order to introduce a number of the features which we attempted to include in our dialogue model. The model itself is not based on an analysis of such dialogues, but rather on a synthesis and extension of existing research in computational generation of dialogues, which we reviewed in the previous chapter. In describing a partial implementation of the model in the next chapter we describe the specific areas of existing research upon which our model was based in detail. *We do not, therefore, claim that our model for tutorial dialogue is psychologically plausible.* In conclusion, we shall argue that these structures are desirable features of intelligent teaching dialogues and that KANT embodies a theoretical model which shows how such structures can be generated in an Intelligent Tutoring System. At the end of chapter 3 we identified a number of trends in Intelligent Tutoring Systems research and research into computational generation of dialogue, many of which correspond closely to our research goals in designing KANT, of incorporating the above characteristics of symmetry, multiplicity of interaction styles, and cooperative negotiation in dialogue. Given these structures and means of analysis, many questions remain:-

- how do the participants decide whether to cooperate or not?
- how are the participant's beliefs updated or revised?
- how do participants decide which goal to pursue next?

and obviously,

- what general mechanisms and knowledge representations are required to generate such

dialogues?

Equipped with these simple theoretical tools and ideas about knowledge representation, in the next chapter we show how KANT generates dialogue structures (but not actual linguistic utterances attached to the structures) with some of the characteristics analysed above. Our hypothetical example represents an ideal in research goals which will enable us to better assess the effectiveness of dialogue traces from KANT which will be examined later. We term the style of tutorial interaction which possesses most of the characteristics analysed above *Negotiated Tutoring*: KANT aims to generate tutorial dialogues of this kind.

Negotiated Tutoring is not identical with critical argument dialogues: it would be possible to substitute a set of dialogue goals which aimed to communicate declarative 'certain' knowledge into KANT, and retain the essential symmetrical and negotiative features of Negotiated Tutoring. However, the requirements of the Negotiated Tutoring approach are particularly apparent in the case where an ITS possesses justified belief, and should therefore not aim to use a highly directive and coercive teaching style in an attempt to simply inculcate these beliefs into the student. The idea of an ITS possessing justified belief lends itself to an egalitarian conception of educational interaction where participants are considered as equals, and are each required to argue critically for their beliefs.

How then do these principal features of Negotiated Tutoring interrelate? It would be possible to represent these features of Negotiated Tutoring as a simple logical deduction from a set of more fundamental and intuitively acceptable premises, making a clear statement which enabled researchers to accept or reject the approach and its consequences on the basis of evidence concerning the effectiveness of teaching approaches and their attendant educational philosophies. We shall state a number of these fundamental principles here, to enable critical assessment, but will not attempt to fully formulate such a deduction. As we showed in the previous chapter, arguments of any kind may be represented in a number of alternative ways (§3.3.2 of this thesis), each of which may reduce or explicate meaning in certain contexts: we believe that for the present, a better way of viewing the intuitive appeal of Negotiated Tutoring is as a set of current approaches and tendencies in ITS research which *naturally cohere*. The first principle of Negotiated Tutoring concerns *interaction symmetry*:

Principle 1

The set of interactional units ('dialogue moves') should be provided for each of the set of dialogue participants.

A second principle specifies a relationship between dialogue level units, and combinations of them, together with a mechanism for shifting topic, into what are commonly termed 'interaction styles':

Principle 2

An interaction style i is equivalent to the set of dialogue moves

{m1, m2, ... mn} and a domain traversal procedure which pairs a set of topics *{t1, t2, ...tn}* with *{m1, m2, ... mn}*.

We recognise that this simple principle will not capture all that is meant by a 'teaching style' in a real educational context, but it is a useful first approximation which admits of refinement in further research.

The final principle relates to *cooperativity*, in terms of the set of principles developed by Grice (1975) with respect to interaction styles, which principle 1 symmetrically quantifies over dialogue participants:

Principle 3

Any interaction style may be pursued if and only if it is negotiated and cooperatively accepted by current participants in the dialogue.

In stating this principle, we are not claiming that this negotiation must be *explicit*.

We must be clear about the theoretical status of these principles: we view them as a reasonably succinct statement of the approach to generating tutorial dialogues which we have described, but since the principles are simply an attempt to describe our model in a clear manner, there is no sense in which they describe the *psychological processes* which underly human multiparty dialogue generation.

From this simple set of principles which place a set of constraints on the interaction, the Negotiated Tutoring approach emerges, with its emphasis on symmetry, multiple interaction styles, negotiation and cooperation. The educative function of such interactions is held to be the facilitation of the development of cognitive skills of engaging in such dialogues, and the effects are presumed to be mutual self-reflection, possible belief revision, and cooperative construction of shared understanding. The full development and formalisation of such a set of principles and their consequences, and the statement of a logical connection with the maxims for cooperative interaction established by Grice (1975) will form a major focus of further research, described in chapter 6 of this thesis.

The next chapter describes a specification and partial implementation in a set of computer programs, of some aspects of the Negotiated Tutoring approach to tutorial interactions. The interaction model is instantiated within additional supporting components of an Intelligent Tutoring System called KANT - the Kritical Argument Negotiated Tutoring system.

Chapter 5: **Specification and implementation of the** **Intelligent Tutoring System KANT**

5.0. Introduction

In the previous chapter we introduced the theoretical constructs required to understand the model of Intelligent Tutoring which we termed *Negotiated Tutoring*. Some aspects of Negotiated Tutoring are implemented in the architecture of the Intelligent Tutoring System 'KANT' (Kritical Argument Negotiated Tutoring System'), which this chapter describes. KANT is partially implemented in Common Lisp (Steele 1984), running on an Apollo Domain® workstation and on Macintosh® computers.

In this chapter we shall describe a prototype implementation of KANT, and specify in detail a number of extensions to the implementation, which do not propose fundamental changes to the architecture within which they are set. We term the union of the detailed description of the prototype implementation, and the extensions to it which are described, the *specification of KANT*, with which this chapter is concerned. Since we aim to fully describe the specification, in a way which would facilitate repetition and extension of this research, the discussion is necessarily detailed. The chapter is structured as follows:

5.0 Introduction and overview of KANT

5.1 The dialogue state

5.2 The dialogue history

5.3 Dialogue goal operators

5.4 An overview of dialogue generation mechanisms

5.5 Some components of the dialogue model in more detail

5.6 Example dialogues and their generation

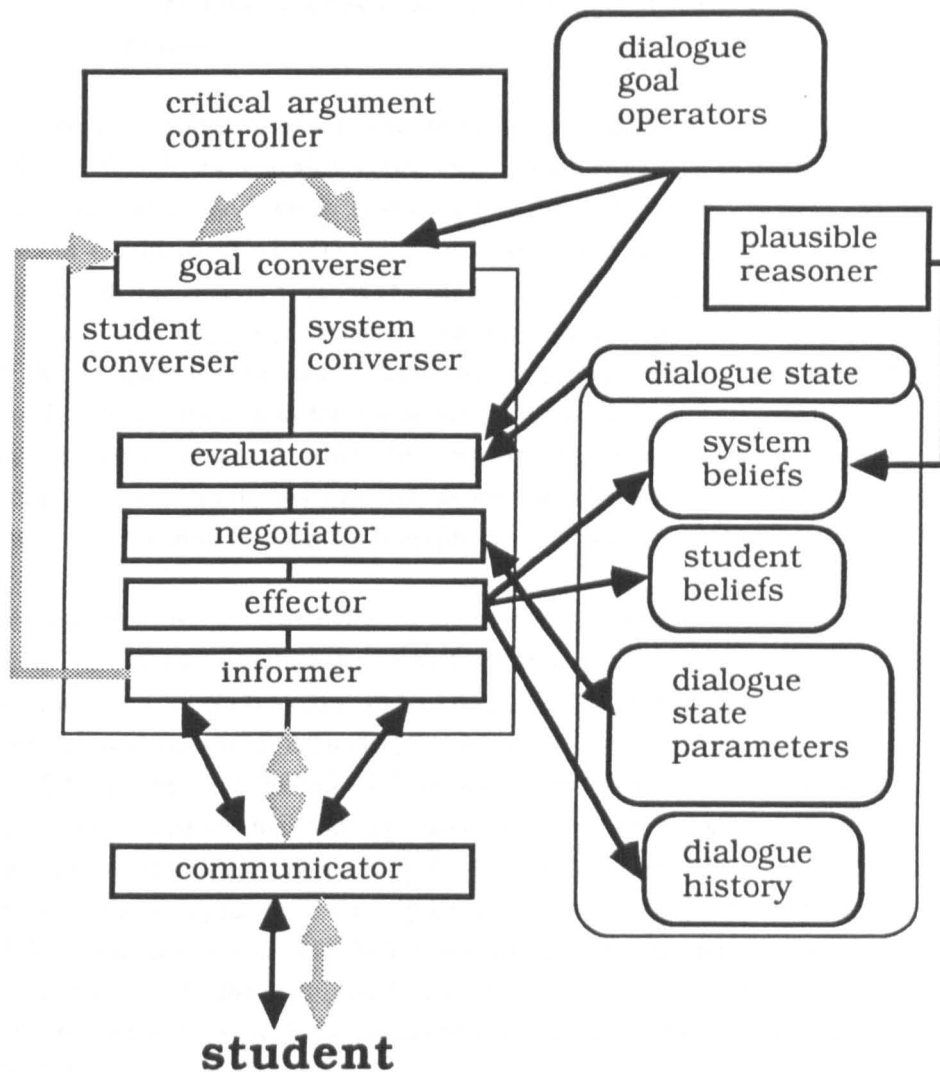
5.7 Extension of the implementation towards the specification

5.8 Summary of chapter 5

5.0.1 An Overview of the Architecture of KANT

Figure 5.0 is a diagram of the architecture of KANT, showing the principal component programs and the data structures on which they operate.

Figure 5.0
The Architecture of KANT



KEY:

□ program

○ data

→ data flow

⇒ control flow

In figure 5.0 square boxes are programs, and rounded boxes are data representations. Bold arrows are data flow, and shaded ones control flow. The architecture represents the knowledge and processes required for the generation of high-level dialogue by a single conversant - *the system* - including mechanisms used to interpret the student's choice of goals and topics to pursue (the STUDENT CONVERSER).

At the highest level, the CRITICAL ARGUMENT CONTROLLER controls alternation of 'negotiated turns' between the conversants, and the decision as to which set of dialogue goals the conversant decides to attempt to negotiate and pursue. Basically, negotiated turns alternate, but not necessarily speaker turns, since a negotiator may prefer the other conversant to be the next speaker.

The principal data structures are a set of possible DIALOGUE GOAL OPERATORS and the DIALOGUE STATE itself, against which the goals are compared to check if they are pursuable in the current state, and which is updated as the dialogue progresses. The dialogue state consists of symmetrical representations of system and student beliefs (a semantic network containing belief instances, a record of dialogue focus, with associated text), a set of parameters for current roles (negotiator, speaker, concept and instance discussed), and a DIALOGUE HISTORY of some instantiated dialogue moves pursued so far. System belief *instances* are assumed to be derived from the frame-based 'plausible reasoner' (described in chapter 2) , which analyses musical structures in an input melody, and returns a set of possible instances of the concept 'phrase_boundary', as well as instances of justification types. These beliefs are represented as instances in nodes for concepts in a representation of long-term semantic memory.

Conversing proceeds in essentially the same manner whether the system or student takes a *negotiated turn*, as part of which conversants are able to propose and negotiate new topics and dialogue moves, cooperative acceptance of which leads to the performance of a dialogue action. The general mechanisms for achieving this are as follows:

- The SYSTEM or STUDENT CONVERSER aims to find a hierarchical succession of dialogue goals which can be satisfied and negotiated in the current dialogue state.
- Goal preconditions are evaluated by the EVALUATOR to check whether the dialogue state preconditions of a goal can currently be satisfied, returning the set of parameter bindings under which this is the case. When the student is negotiating, the system evaluates preconditions of the proposed dialogue goal with respect to the dialogue state, when substituted with the parameters for 'concept', 'instance' and 'speaker' roles.
- The NEGOTIATOR manages the negotiation phase concerning these satisfiable parameter bindings, calling the COMMUNICATOR where necessary, and returning the negotiated and cooperatively agreed parameter bindings for use in the INFORMER and EFFECTOR.
- The INFORMER performs dialogue actions, using text generation templates. When the student is conversing it records and checks the beliefs which the student is communicating. The INFORMER can recursively call the CONVERSER as part of INFORMING the student about some concept (teaching required subconcepts for understanding the higher level dialogue action), and uses the COMMUNICATOR to actually generate text.
- The EFFECTOR propagates the effects of dialogue actions on the dialogue state, thus updating the belief models and dialogue history.
- The COMMUNICATOR handles the (text-template) input and output involved in negotiation and performance of dialogue actions. It contains separate text templates for dialogue actions, and for metadialogue utterances involved in negotiation, and generation of

preannouncements involved in informing the student about precursory knowledge.

Let us explain these components in a little more detail, along with the generation algorithm.

5.1 Representing the Dialogue State

In KANT the 'dialogue state' represents the changing course of participants' beliefs and semantic knowledge, concerning the domain of discourse, the beliefs of the other participant, and the state of the dialogue itself (the dialogue history and the current set of dialogue state parameters). KANT represents a dialogue state for the system, and we assume that the student possesses similar mental states and processes. In section 5.1 we therefore describe:

- the notion of a dialogue state ;
- the domain knowledge and belief representation for the system, and its representation of the student's knowledge and beliefs ; and
- the theory of semantic memory on which the knowledge representation is based.

5.1.1 The notion of a 'dialogue state'

The notion of a "dialogue state" was first described by Power (1979)¹:

"... the dialogue state is the situation which has arisen between two speakers as a result of their previous dialogue."

(p. 131)

In his program which simulated cooperative planning between two hypothetical robots

"... a robot represents his dialogue state by his control stack in conjunction with his copy of the planning tree."

(p.131),

in other words, the portion of the current plan (with parameters for speaker roles instantiated), and the planning tree. We extend this notion of the dialogue state to include parameters for the current topic of conversation ("c") and for the current negotiator ("n"), separable from the proposed speaker in the dialogue goal proposed for negotiation. This is required from our concern with goal operators for *dialogue* (as opposed to physical) goals, and our isolation of the explicit negotiation phase as separable from dialogue actions and the speaker role who performs them. Our conception of "the situation which has arisen between two speakers as a result of their previous dialogue" therefore includes :

- the portion of speakers' beliefs arising from possible belief revision in the dialogue ;
- the portion of speakers' beliefs concerning the other speaker's beliefs, abstracted from the previous dialogue ;
- the speakers' current proposed values for dialogue parameters (negotiator, speaker, concept, instance) ; and
- the speakers' memory of dialogue actions (and their parameter bindings) performed so far.

¹See chapter 3, §3.4.5.1 of this thesis for further discussion.

We isolate these memory representations as the 'dialogue state' since they change during the course of the dialogue, and they form the knowledge and beliefs which participants use to decide how to continue the dialogue.

In the current version of the theory, the set of possible dialogue goals is assumed invariant (as in Power's 'planning tree'), and symmetrically available to either conversant: it is therefore not viewed as part of the dialogue state, but rather of long-term memory. Our notion of 'dialogue state' is therefore fundamentally that portion of memory (a working memory) which influences participants' decisions in dialogue, and which changes dynamically in the course of interaction, thus 'driving the dialogue forward'.

5.1.2 The Domain knowledge and its Derivation

The knowledge of the domain of discourse possessed by a dialogue participant must derive from a combination of new knowledge derived from the current dialogue, with existing knowledge in semantic memory (Quillan 1968; Collins & Loftus 1975). In KANT, this new knowledge is assumed to be derived from the frame-based plausible reasoner for analysis of musical structures, described in chapter 2 (since the integration of these two components was not completed). The output of the reasoner would be a set of justified beliefs, being the values filled into slot types for schemata concerning possible overall musical structures: these are the specific instances of concepts and of justifications for belief in instances of these concepts that the system possesses. Given the large degree of research effort involved in creating a reasoning model, we fundamentally consider a single belief type: 'phrase boundary'. The following is an example of a set of such beliefs which could be output by the parser²:

```
(belief_type= phrase_boundary)
(belief_instances= (
  (belief_position= (notes= (4 5)))
    (justifications=
      (opening_progression
       interval_leap_contrast
       rhythm_parallelism)))
  (belief_position= (notes= (10 11)))
    (justifications=
      (closing_progression)))
```

In this example, there are two instances of the concept 'phrase_boundary' believed by the system, marked as positions (4 5) and (10 11), referring to the position of the phrase boundary in the melody, between the 4th and 5th and 10th and 11th notes from the beginning, respectively. We can envisage the tutorial situation where a teacher has the teaching material of a melody displaying two instances of the concept 'phrase boundary', which (s)he wants to teach, and the teaching problem is to use these instances or examples to form a coherent tutorial dialogue

² Since the frame-based reasoner was not fully implemented, we have described it in sufficient detail to be able to specify its hypothetical output as input to KANT (see chapter 2).

which teaches the concepts involved, and any precursory knowledge required. Any reader not familiar with this musical domain will realise that to construct an interaction around simply stating or 'critiquing' (Miller 1984) justifications, would be uninformative to a dialogue participant who was not familiar with these kinds of justifications: we need to produce coherent dialogues which integrate dialogues which have variously been termed argument, explanation, instruction, critiquing, and so on. In the case of justifications for postulating a phrase boundary in tonal music, it is important to remember (as we argued in chapter 2) that there are no sufficient conditions: beliefs of this kind are therefore inherently uncertain, and can not be simply communicated to students with the aim of transferring these beliefs.

5.1.3 System and student belief representations

In this section we discuss the representation of semantic knowledge and belief used for the system and its representation of the student's mental states, together with its relationship with existing theories of semantic memory.

5.1.3.1 The semantic network representation

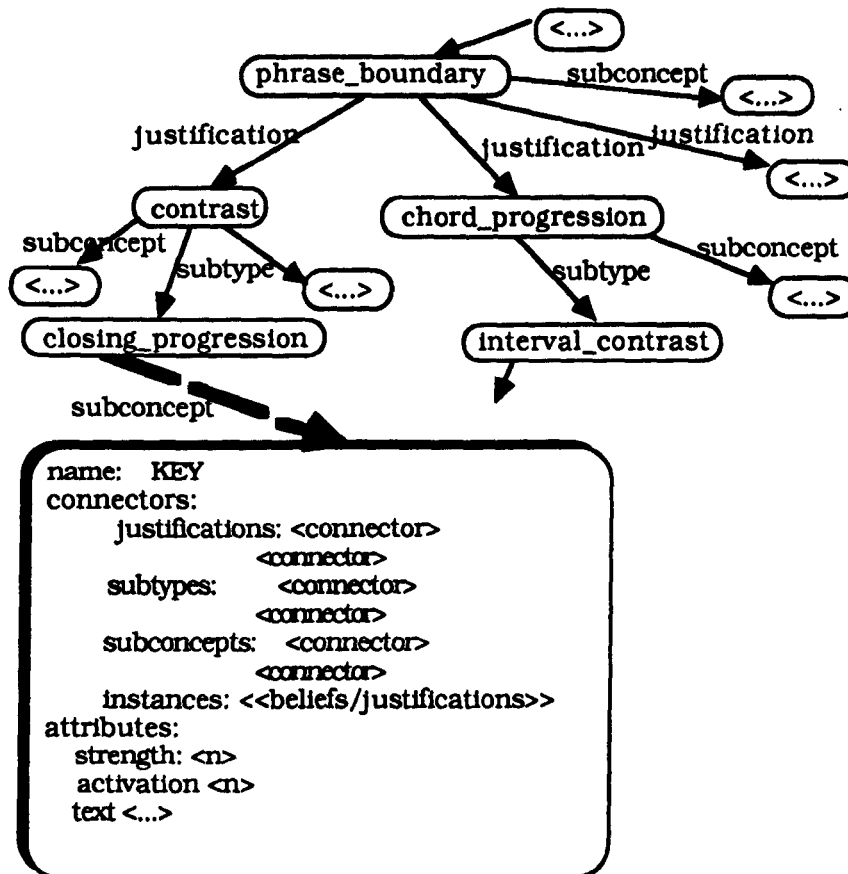
The system's knowledge of specific instances of concepts is integrated with a model of semantic knowledge which interrelates sources of justifying knowledge, represented in the form of a semantic network (Quillan 1968). Thus, for example, the system's belief that $p1$ is a phrase boundary, with justifications $j1$ and $j2$ would be represented as the presence of ' $p1(j1\ j2)$ ' as a value in the 'has_instances' attribute of the node 'phrase_boundary' in the semantic network, where $j1$ and $j2$ are justification types, attached as nodes linked by a 'has_justification' link-type to 'phrase_boundary'. Concept nodes have all the same format, representing long or short-term memory traces, encoded in the course of the present or a previous dialogue. Nodes have a consistent representation, whether they are justifications, subconcepts or subtypes. A portion of the network is shown in figure 5.1, and a complete network for teaching musical phrase boundaries is listed in Appendix A. All nodes have:

- a concept name ;
- links for
 - 'has_justification',
 - 'has_subconcept',
 - 'has_subtype',
 - 'has_instance',

and their inverses (for example, the inverse of "has_subconcept" is "subconcept_of") ; and

- attributes for
 - node strength,
 - activation level, and
 - explanatory text .

Figure 5.1
A portion of the semantic network linking sources of justifying knowledge



Quillian (1968) proposed semantic networks as a formalism for representing conceptual knowledge in semantic memory, and proposed 'spreading activation' as a mechanism for searching such networks in memory retrieval. The classic problem considered was to account for timing differences between judgments of similarity or difference between pairs of words (eg CRY/ COMFORT), where two concepts can be said to be similar when activation spreads by association from each concept in memory, and finds an intersection which expresses a conceptual relationship: the greater the recall time, the less related are the two concepts. Collins & Loftus (1975) proposed the following extensions and clarifications of Quillian's paper:

- (1) When a concept is stimulated, activation spreads out along the paths of the network in a decreasing gradient which is inversely proportional to the strength of the links in the path.
- (2) The longer a concept is continuously processed (reading, rehearsing etc) the longer activation is released from the node of the concept at a fixed rate. Only one concept can be actively processed at a time, due to serial nature of cognition.
- (3) Activation decreases over time and/or intervening activity. Activation can only start out at one node at a time.

(4) Intersection requires a threshold for firing. Activation from different sources summates , and when it reaches a threshold , the path in the network producing the intersection will be evaluated.

(5) The semantic network is organised along lines of semantic similarity, based on an aggregate of the interconnections between two concepts. The more properties two concepts have in common, the more links there are between the two nodes via these properties and the more closely related are the concepts.

We have based the semantic memory representations for dialogue participants in KANT on a later theory which incorporates a model of spreading activation, developed by Anderson (1983). This version of the theory is sufficiently formal to be applied in the development of an ITS. It provides us with a model of *encoding* of memory traces for topics explicitly mentioned in dialogue, and a partial theory of *dialogue focus*, in terms of activation which spreads from concepts entering working memory in the course of a dialogue (either from long-term memory, or newly encoded concepts which have been mentioned).

5.1.3.2 Anderson's theory of factual memory

Anderson (1983) incorporated most of the elements discussed by Collins & Loftus (1975) into the version of spreading activation theory used in his 'ACT' theory, which is the model of memory encoding, retention and retrieval used in KANT. In order to understand fully the way in which this theory influences memory encoding and dialogue focus models in KANT, we need to understand Anderson's version of the theory in some detail. In Anderson's theory, information is encoded in an all-or-nothing manner in working memory into a network of 'cognitive units', the strength of which increases with mental rehearsal and decays with delay. Transient copies of cognitive units are created in working memory, where the basic encoding hypothesis of ACT is that there is a probability that a working memory cognitive unit will become a permanent long-term memory *trace*. This probability does not vary with intention and motivation to learn, nor with duration in working memory; study time has little effect when information is not being actively processed, and probability increases with repetition. This gives a very simple theory of additional learning:

- when created, a trace has an associated strength of 1 unit ;
- each subsequent trial increases strength by 1 unit ;
- overlearning increases speed of retrieval and probability of retention.

In terms of retention, the strength of a trace decays with time according to the simple formula

$$s = t^{-b}$$

where

s = trace strength

t = time measured from point at which trace is created in working memory

b = has value $0 \leq b \leq 1$

So the total strength of a trace is simply the sum of multiple strengthenings.

Spreading activation is the mechanism proposed for memory retrieval. Working memory is equal to the information currently available for processing. It contains encoded information, inferences, goal information and long term memory traces .

The continuous nature of spreading activation means that membership of working memory is therefore a matter of degree. At any point in time, certain working memory elements are sources of activation (eg being encoded, or internal concepts currently being processed). Activation spreads from these activated units to associated elements. As soon as a source drops from attention, its activation begins to decay, as does activation in long term memory. A pattern of activation in long term memory is therefore set up in response to the input of activation from the focused units. If a node n_y receives activation a_{1y} to a_{iy} , from nodes n_1 - n_i , its level of activation is simply the sum of these input activations. Activation sent by node n_x to nodes n_1 - n_j is determined by strength s_1 to s_j and the activation level of node n_x . If we have a network of m nodes we have m simultaneous equations with variables a_1 - a_m , which can be solved to find the complete activation pattern set up by a particular set of focussed elements in working memory. Note that this scheme allows activation to reverberate back: i.e. if node 1 connects to node 2, activation will spread to node 2 and activation from node 2 will spread back to node 1. We can, however, make a number of approximating assumptions:

- (1) "... the impact of distant structure is minimal and one's derivations will be quite accurate assuming only the proximal network structure." (Anderson 1983, in Collins & Smith 1988, p.139)
- (2) "... assume that activation will only flow forward from the working memory elements and not reverberate backwards." (op cit, p.139)

The simplified formula for spreading activation (which is the one used in KANT) now becomes as follows:

the activation that node n_x sends to nodes n_1 to n_j is determined by the strength s_1 to s_j of each node and the activation level of node n_x .

If its level of activation is a_x , the amount of activation it sends to node n_k is

$$a_x s_k / \sum_i s_j$$

where l is the loss in activation and $l a_x s_k / \sum_i s_j$ is the relative strength of node n_k from n_x .

5.1.3.3 Use of Anderson's theory in KANT

Anderson's theory of memory is used in KANT, in terms of both encoding and retrieval. It is clear that in dialogue, participants use some kind of working memory for concepts mentioned and for remembering what has been said so far. We can therefore say that Anderson's model provides a mechanism for encoding, additive learning (or 'remembering'), and for shifts in dialogue focus as new concepts are brought into working memory by spreading activation from previous dialogue foci. In summary, therefore, Anderson's memory model is used in KANT in the following ways:

- (1) When a concept is explicitly mentioned in the dialogue by either participant, we assume that any participant who does not already possess this concept node, encodes it. In literal terms, if the system mentions the concept 'chord', for example, then if the student does not already possess a node for this concept (in terms of the system's identical representation of the student's beliefs) then a copy of the node for 'chord' from the system's network is inserted into its representation of the student's beliefs.
- (2) KANT lacks a perspective concerning the origin of the participants' long-term memories over time. For this reason we make the simplifying assumption that all nodes (those which are assumed as the initial memory of the system, and those which are encoded during the dialogue) have an initial node-strength of one unit.
- (3) As with Anderson's theory, each time a concept is explicitly mentioned by either dialogue participant, we assume that the strength of the appropriate concept node is increased in the long-term memory of each by one unit, since it has been mutually mentally rehearsed.
- (4) Dialogue focus is recorded in the network as current activation level of a concept node. When a concept is explicitly mentioned, the appropriate concept node in the semantic network representation of each participant becomes a source of activation of input = 1. The units involved are of no importance since what is important is the *relative* activation of each node. This has the functions of enabling the system to choose the next focus (concept) to discuss, and of enabling the system to decide whether to cooperate with a new focus proposed and negotiated by the student: the next focus is the node (non-identical with the previous focus) which has the highest activation level (given certain other conditions, such as the fact that the student does not already understand this concept, and so on), and the system will cooperate with a proposed focus provided it has sufficient activation. We shall discuss the details of exactly when and how focus is chosen and updated in the later sections on 'action effects' and 'dialogue state preconditions', since focus changes are effects of the 'dialogue actions' of stating a belief about some concept, and focus level is a precondition for discussing some concept. Since the student and system have separate belief representations, we have a separate model for the system's view of current focus and the student's view of it.
- (5) Finally, we assume that activation decays with time, according to Anderson's equation above. 'Time' here does not mean real-time, measured in seconds, since time of execution of the Lisp functions in which KANT is implemented will vary with each machine and programming environment. Instead we use 'dialogue unit time' as a measure of the age of encoding of a memory trace (concept node), where one unit elapses after each *dialogue action* is performed by either participant.

The explicit mentioning of some concept in the dialogue, as part of a 'dialogue move' enables either participant to encode a representation for that concept, and for that representation to be an input source for 'spreading activation' according to Anderson's (1983) version of that theory.

5.2 The dialogue history

The possibility of a speaker challenging their own previous claims has the quite important consequence that KANT does not have to maintain a 'goal stack' of utterances which have not been completed - a speaker can simply return to an uncompleted previously chosen sequence of goals, and attempt to resatisfy dialogue state preconditions for the utterance of those goals. The following example of a dialogue history value after a short interaction should make this clear, where the *first* element in the list ('move 1') is the most recent move (numbering is for convenience only):

[illegible]

```

(S= 'SYSTEM)
(INST= 'P1)
( (GOAL_NAME= 'SUPPORT_INSTANCE_CLAIM) #|move 6|#
  (S= 'STUDENT)
  (C= 'PHRASE_BOUNDARY)
  (INST=(CONTRAST CHORD_PROGRESSION)))
( (GOAL_NAME= 'MAKE_INSTANCE_CLAIM)           #|move 7|#
  (C= 'PHRASE_BOUNDARY)
  (INST= 'P1))
( (GOAL_NAME= 'CONCRETE_CLAIM)                 #|move 8|#
  (S= 'STUDENT))
( (GOAL_NAME= 'CLAIM)                           #|move 9|#
  (S= 'STUDENT)
  (C= 'PHRASE_BOUNDARY)
  (INST= nil))

```

The dialogue history is a simple list of dialogue moves pursued and the new parameter bindings made for that move (if from one move to the next there is no recorded value for the speaker - *s* - then the binding has remained the same as in the previous move). The dialogue fragment *begins* at

move 9:

the student has successfully negotiated that a dialogue move 'claim' ('goal_name= 'claim') will be pursued, with the student as next speaker ('s= 'student'), concerning the concept 'phrase_boundary' ('c= 'phrase_boundary') - at this stage the value for the instance of this concept to be discussed remains unbound ('inst= nil').

move 8:

student successfully negotiates goal_name= CONCRETE_CLAIM, as opposed to the making of an abstract_claim.

move 7:

now a move is successfully negotiated which has a dialogue action attached, and the instance is bound to p1 (i.e. the student has stated that p1 is an instance of a phrase boundary).

move 6:

the student states the support types for the claim, being the existence of a musical 'CONTRAST', and the type of 'CHORD_PROGRESSION' found at the claimed phrase boundary.

move 5:

This move corresponds with a change of negotiated turn to the system, where preconditions for challenging the previous claim about the instance p1 of the concept 'phrase_boundary' can be met.

move 4:

The system states that it agrees that p1 is an instance of the concept.

move 3:

The system agrees with the previously stated support type 'CONTRAST'

move 2:

The system states its belief that 'PARALLELISM' is a new instance of a support type for p1.

move 1:

Finally, the system disagrees with the previously stated support type 'CHORD_PROGRESSION' (i.e. it does not share this belief).

In summary, the dialogue history performs the function of enabling the system to refer dialogue moves to those which have already been stated by itself or the responding student.

5.3 Dialogue goal operators

In a conversation between two speakers, whilst one is speaking the other generally listens, attempts to understand, and more or less consciously decides what to say next. Their response - if they are given opportunity to respond - may consist of a simple acknowledgement of what has been said, and an invitation for the first speaker to go on, it may consist of a detailed response to what has been said, with acknowledgement of the connection of what is to be said with what has been said, it may consist of an acknowledgement and the making of a fresh statement on a related topic, or a number of other possibilities. In a genuinely interactive dialogue, a participant will generally have to 'think on his feet', and choose quickly between these possibilities. Where each participant is given sufficient freedom, and neither explicitly or covertly controls the interaction, it will generally be difficult for either to adhere to a preconceived 'plan' for the direction of the future dialogue: they will choose the 'best' of the available possibilities available to them in the current state of the dialogue, for guiding the dialogue in general terms. It may be, for example, that they prefer to remain silent and listen to what the other says, and any number of other general preferences.

A computational model of dialogue must therefore have some representation of the various conversational possibilities, and some way of choosing between conversational possibilities at any point in the dialogue. We shall deal with the mechanism for choosing what to say next, and how to say it, in the next section, and describe the representation in KANT used for a set of possible conversational moves available to both participants in the tutorial dialogue.

In KANT, what we have termed 'conversational possibilities' (Grice 1975), are represented as special kinds of goal operators, which are termed *dialogue moves*. Each conversant has access to approximately the same set of dialogue moves³, and any dialogue move can be 'instantiated' with a number of values for parameters concerning

n = the current negotiator

s = the next speaker proposed by n

c = the next topic (concept) proposed by n

i = the next instance of c proposed by n

The set of available dialogue moves is extremely influential in controlling the interaction, since nearly all the features generally associated with Intelligent Tutoring Systems, are

³ As we describe later, an important restriction of the system is that the student can not make unrestricted explanations for new concepts (the 'ABSTRACT_CLAIM' move).

controlled by procedures attached to them. For example,

student modelling

The model of the student's beliefs is updated as an *effect* of a goal action, i.e. the actual performance of a dialogue action which communicates some belief.

choosing the next interaction step

The next interaction is chosen by determining which dialogue moves are satisfiable in the current dialogue state, with respect to their *preconditions*, choosing between these alternatives according to a set of general educational preferences, and attempting to negotiate the preferred interaction step.

generating a teaching interaction

A teaching interaction corresponds to a 'dialogue action' (a text template attached to a concept node), performed when the preconditions for the action are satisfied with respect to the dialogue state and cooperative negotiation.

5.3.1 The general structure of dialogue moves

Let us now look at some example goal operators corresponding to dialogue moves. Figure 5.2 shows the form of a single dialogue goal.

Figure 5.2
Example dialogue goal operator for making a CLAIM

```
(CLAIM
(c inst n s)
(dialogue_state_preconditions=
  (and
    (not (null c))
    (OR
      (null *dialogue_history*)
      (AND
        (in_focus? c (ltm_model s))
        (not (known? c (ltm_model (dialogue_participant s))))
        (exists_ltm_trace? c (ltm_model s))))))
(negotiation_preconditions=
  ((negotiate s (goal= 'CLAIM))
   (negotiate s c)))
(subgoals=
  (OR ((concrete_claim (c inst s))
        (abstract_claim (c inst s)))))
(negotiation_effects=
  ((update_dialogue_history
```

```

'((goal_name= 'CLAIM) (s= s) (c= c) (inst= inst))))
(action_effects= nil)
(actions= nil))

```

The dialogue goal operators used in the first version of KANT were extremely simple, since they were intended to demonstrate that the proposed approach to tutorial dialogue generation was feasible. The complete set used is listed in Appendix B. In figure 5.2 the example operator has eight components, marked in bold typeface on the example:

- (1) a goal name - 'CLAIM' ;
- (2) a set of parameters (c inst n s) ;
- (3) a set of dialogue state preconditions ;
- (4) a set of negotiation preconditions ;
- (5) a set of subgoals ;
- (6) negotiation effects ;
- (7) action effects ;
- (8) actions.

We shall discuss each type of component for all goals in later sections; but for now we shall briefly describe each component in the example.

(1) *goal name*

The goal name is to make a 'CLAIM' about some belief. This is understood in very general terms at this stage in a hierarchical set of goals.

(2) *goal parameters*

These parameters are bound to certain values in the process of dialogue generation. The parameters may be initially bound, in whole or in part, during interpretation of some higher level goal, and generally speaking, bindings are passed down the hierarchy. A typical binding might be

```

c    <= harmony
inst <= i10
n    <= system
s    <= student

```

which means that the system is currently trying to negotiate that the student should be the next speaker about instance i10 of concept 'harmony'. As part of the process of attempting to converse this goal, these bindings are substituted throughout the dialogue state preconditions. All parameters do not have to be bound, since in some cases, as part of the process of evaluating dialogue state preconditions, new parameter bindings are found under which the preconditions could be satisfied - for example, in the goal 'concrete_claim', if 'c' is initially unbound, then a special procedure attempts to find a binding under which a concrete claim could conceivably be made at that point. This is marked by a special procedure called 'bind', which simply returns the value of its argument - for example

```

(bind 'c (get_concept_to_discuss *system_ltm* *student_ltm*))

```

returns the binding of 'c' to the value returned by the function 'get_concept_to_discuss', which may of course be nil, in which case this precondition will fail (i.e. you cannot satisfy a goal of making a claim if no concept can be found concerning which you could reasonably make a claim).

(3) *dialogue state preconditions*

The set of dialogue state preconditions is a simple *logical expression*, represented in terms of Lisp functions. Their interpretation is very simple, and uses the usual semantics of 'and', 'or' and 'if', as defined in Lisp predicates:

- if the complete expression returns the value 'true', then the preconditions are said to be satisfied, otherwise they are unsatisfied ;
- a returned value of 'true' is the value 't' or any non-nil value ;
- the satisfied expression returns the parameter bindings under which it was satisfiable ;
- if new bindings are produced (when the 'bind' function is encountered), then the set of parameter bindings returned reflects these new bindings :
- to determine whether the preconditions are satisfied, parameter bindings are substituted for the relevant expressions, and the substituted expression is simply *evaluated* in the current Lisp environment (which includes a global representation of the dialogue state).

The explanation for our simple example set of dialogue state preconditions is therefore as follows:

```
(and
  (not (null 'harmony)) #|expression 1|#
  (OR
    (null *dialogue_history*) #|expression 2|#
    (AND
      (in_focus? 'harmony (ltm_model 'student)) #|expression 3|#
      (not (known? 'harmony #|expression 4|#
        (ltm_model
          (dialogue_participant 'student))))
      (exists_ltm_trace? #|expression 5|#
        'harmony (ltm_model 'student))))))
```

The overall expression is an 'and', so both expressions 1 and the value of the 'or' expression must evaluate to 'true'.

#|expression 1|#

the c parameter must now have a non-nil binding

#|expression 2|#

evaluates to true if the dialogue history is nil - i.e. try to make a claim if it is the start of the dialogue

#|expression 3|#

if the student is to be the next speaker, then the concept 'harmony' must be in focus from the student's point of view

#|expression 4|#

and, the respondent (the system) must not already know or understand that concept, if we are to decide to tell them about it

#|expression 5|#

This simply says that for a speaker to make a claim about a concept, they must have a memory-trace (concept node) for that concept.

(4) *negotiation preconditions*

If the composite dialogue state precondition expression evaluates to true, then the parameter bindings are returned, and passed to the negotiation preconditions, which are now substituted with these bindings. In our example, we would now have a simple list of substituted expressions:

```
(negotiation_preconditions=
  ((negotiate 'STUDENT (goal= 'CLAIM))
   (negotiate 'STUDENT 'HARMONY)))
```

The general evaluation rules are again very simple:

- if all negotiation preconditions succeed then negotiation succeeds, otherwise negotiation fails ;
- each precondition is evaluated with respect to the current Lisp environment, evaluating the function 'negotiate' with appropriate arguments ;
- if the negotiation preconditions succeed, the goal succeeds, otherwise it fails.

If and only if a goal succeeds, then actions can be performed, effects propagated on the dialogue state, and subgoals be pursued. Negotiation may result in *new* parameter bindings, as a result of attempting to resatisfy dialogue state preconditions with parameters proposed during the negotiation process. We will discuss the details of the negotiation process in a later section.

(5) *subgoals*

Subgoals inherit the final parameter bindings produced after successful negotiation. They consist of a list of subgoals, prefixed by one of the logical operators

'OR' - exclusive subgoals

'AND-OR' - one or more subgoals may be pursued and satisfied

'AND' - all subgoals must be satisfied

(6) *negotiation effects*

Effects of successful negotiation are that a function is evaluated after substitution with final successfully negotiated parameter bindings. This function simply updates the dialogue history:

```
((update_dialogue_history
  '( (goal_name= 'CLAIM)
    (s= 'STUDENT)
    (c= 'HARMONY)
    (inst= NIL))))
```

(7) action effects

These operate to update belief models as the result of a dialogue action, again by the evaluation of appropriate substituted functions. In this case we have no actions attached, and hence no action effects. A typical example for goals which might have actions would be:

```
(action_effects=
  ((update_ltm_models '((c= 'HARMONY) (inst= 'i66))))))
```

which would update both models in the manner described in the earlier section on use of Anderson's spreading activation model.

(8) actions

Our example has no actions. Actions involve evaluating a substituted version of the function 'INFORM', which generates the appropriate text template attached to a concept node (in the case of the system as speaker), or a statement which summarises the goal and topic (in the case of the student as speaker). An example is:

```
(actions= (INFORM 'STUDENT ('HARMONY 'i666))))
```

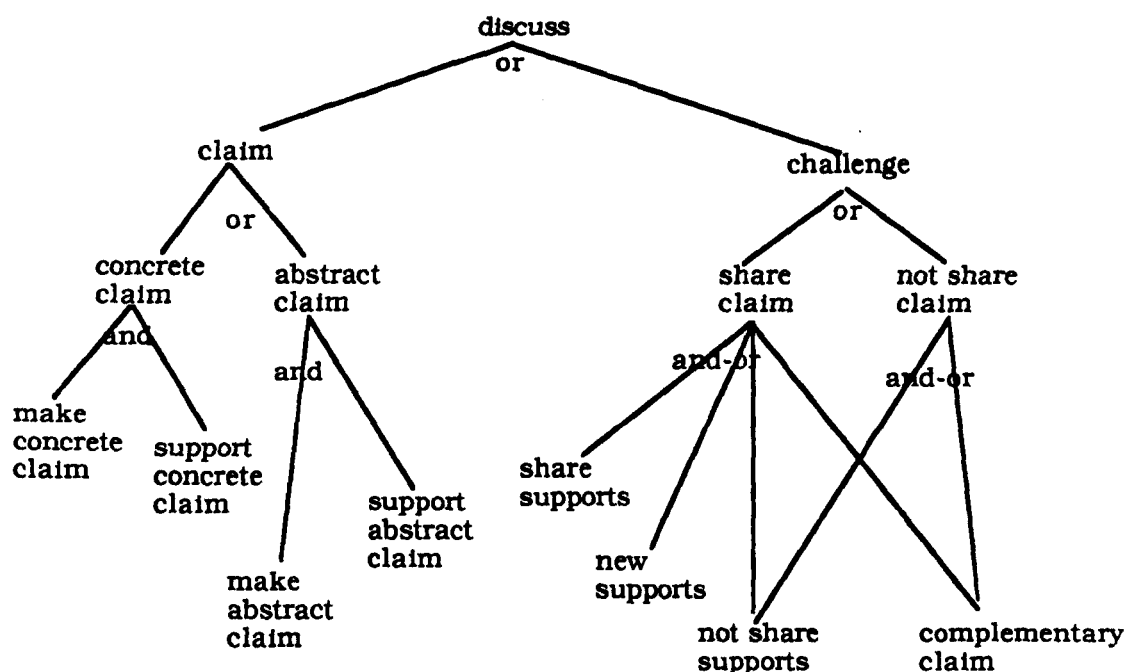
in which the student informs the system that 'i666' is an instance of the concept 'HARMONY'. 'Informing' is a reasonably complex procedure, which may involve recursive application of the complete dialogue generation mechanism on subconcepts of the initial concept informed. We shall therefore discuss it in more detail in a later section.

From the examples which we have given, it is apparent even at this stage in our description of KANT that dialogue goal operators are the central representation which controls dialogue generation through attached procedures and logical expressions, which are substituted with values bound to parameters for speaker, concept and instance roles.

5.3.2 The hierarchy of dialogue moves

Let us now examine the range of dialogue goals provided in KANT. Figure 5.3 shows a simplified hierarchy of possible goals.

Figure 5.3
A simplified dialogue goal hierarchy



The goal tree is *hierarchical*, in the sense that higher level parent goals express more abstract or general types of dialogue move than their children. Accordingly, higher level moves have more generalised dialogue state preconditions, as we saw in the earlier example. A fundamental distinction is made between dialogue goals which involve the making of some type of CLAIM, and those which involve CHALLENGING a claim which has already been made, which are of course mutually exclusive in a single negotiated turn. This captures the intuition that a respondent may decide either to state something new, or to respond in detail to the conversational initiator's utterance. The highest level goal is to DISCUSS some topic, which will always be satisfied provided that conditions for terminating the dialogue are not met - i.e. neither the system nor student wants to voluntarily terminate the dialogue.

5.3.2.1 Making claims

Making a claim generally involves stating some belief which will be related to previous utterances in terms of topical connections, but which does not directly refer to a previous utterance. At the next level of abstraction, CLAIM goals divide into either *abstract* or *concrete* claims. An abstract claim corresponds to something like an 'explanation', and involves making some general statement about the meaning of a concept, without actually stating a concrete instance. In implementational terms, KANT simply generates the explanatory text associated with a concept node. Conjunctive subgoals for abstract claims involve the actual making of that claim (MAKE_ABSTRACT_CLAIM has an attached dialogue action), and a general statement about the kinds of justification types which are relevant to instances of such claims (SUPPORT_ABSTRACT_CLAIM). It is of course possible that in a full implementation

(rather than one simply intended to demonstrate the feasibility of an approach), that a further set of subgoals could be represented here which have been identified as dialogue moves in explanation dialogues (such as in the work of Weiner 1980).

Concrete claims involve making an 'illustration' of a concept by describing a specific instance and its specific justifications. With respect to discussion of a specific topic, this is related to Reichman's (1985) distinction between 'instance' and 'issue' context spaces. The subgoals are simply to make that specific instance claim, and to state the supports for it. In the goal tree, these subgoals are marked as *conjunctive*, which means that the conversant (system or user) is committed to producing supports for any claim if they wish to make a claim in the first place. This is one way of avoiding the necessity to *elicit* supports on this branch of the tree. All other alternative goals are disjunctive in a *non-exclusive* sense (i.e. any combination of the subgoals may be chosen, in a specific ordering from left to right, for tutorial reasons).

5.3.2.2 Making challenges

On the CHALLENGE branch of the goal tree, the goals divide into whether the current conversant agrees with, or disagrees with, the claim which is to be challenged. In the case of a participant challenging a claim which they themselves have previously made, with which they presently disagree, (according to the dialogue history), this corresponds to having revised their beliefs. An interesting property of the subgoals for challenging a claim, with either agreement or disagreement, is that the subgoals are exactly the same in each case: they differ only in their *ordering*. There are intuitively acceptable reasons for this. In critical argument conducted with a view to didactic purposes, one would wish to examine a student's reasons for making some claim whether one agreed with the claim or not. Secondly, in human critical argument where one adopts an *adversarial* position⁴, it seems most natural to begin by discussing those points on which one disagrees initially, and move on to points of agreement afterwards (or else assume that what is unsaid is agreed). The converse seems to be true in situations where one shares a conversant's beliefs. Weiner (1980) has discussed these issues with respect to generation of explanations from expert systems. For these reasons we have maintained a distinction between agreeing and disagreeing with a claim in challenging. The explanation of these subgoals is quite straightforward:

a conversant states those justifications with which (s)he

- agrees (AGREE_SUPPORTS),
- with which (s)he disagrees (DISAGREE_SUPPORTS) and
- in cases where a conversant believes that there is some possible justification which the respondent has not stated, (s)he states these new supports (NEW_SUPPORTS).

In cases where the student has stated that a set of justifications justify an instance of a concept at some position, the system would look for other positions in the melody (perhaps already encapsulated in its own belief set) where the same justification set also holds, and ask the student if they would also consider this to be an instance of the concept

⁴ We understand the term "adversarial" here simply in the sense of 'disagreement'. The system does not support adversarial arguments in the true sense, since we assume cooperativity in the dialogue model.

(COMPLEMENTARY_CLAIM), possibly leading to a revision of their beliefs. In a knowledge domain where certain beliefs could be shown to logically or practically contradict each other, an adversary in a critical argument would clearly want to point this out. In our domain there are no such strictly logical contradictions, since we are concerned with justified beliefs derived from a plausible reasoning mechanism (chapter 2 of this thesis).

5.3.3 Dialogue moves and teaching styles

Using this small number of very general dialogue goals, we can represent a reasonably large range of multiple interaction styles, which are available to both the system and the student. Each goal is represented in a completely general way so it can be pursued by either participant, which effectively doubles the range of possible interaction goals. Let us consider some examples of combinations of dialogue goals representing teaching styles, where we write the parameters for a goal as follows

```
goal_name(negotiator, speaker, concept, instance)
n = negotiator
r = respondent, i.e. non-negotiator
```

In order to understand how these dialogue moves relate to conventional teaching styles, we shall assign some names to some examples of sequences of such goals, with specific bindings:

```
(1) SYSTEM_INFORM
["system informs concrete belief instance"]
discuss(system,_,_)
claim(system, system,c1,_)
concrete_claim(system, system,c1,i1)
make_concrete_claim(system, system,c1,i1)
support_concrete_claim(system, system,c1,(s1,s2))

(2) STUDENT_INFORM
["student informs concrete belief instance"]
discuss(student,_,_)
claim(student, student,c1,_)
concrete_claim(student, student,c1,i1)
make_concrete_claim(student, student,c1,i1)
support_concrete_claim(student, student,c1,(s1,s2))

(3) SYSTEM_ELICIT
["system requests STUDENT_INFORM"]
discuss(system,_,_)
claim(system, student,c1,_)
concrete_claim(system, student,c1,i1)
make_concrete_claim(system, student,c1,i1)
support_concrete_claim(system, student,c1,(s1,s2))

(4) STUDENT_ELICIT
["student requests SYSTEM_INFORM"]
discuss(student,_,_)
claim(student, system,c1,_)
concrete_claim(student, system,c1,i1)
```



```
make_concrete_claim(student, system,c1,i1)
support_concrete_claim(student, system,c1,(s1,s2))
```

(5) SYSTEM_EXPLAIN

```
["system makes informs general concept explanation"]
discuss(system,_,_)
claim(system, system,c1,_)
abstract_claim(system, system,c1,i1)
make_abstract_claim(system, system,c1,i1)
support_abstract_claim(system, system,c1,(s1,s2))
```

(6) STUDENT_EXPLAIN

```
["student makes general concept explanation"]
discuss(student,_,_)
claim(student, student,c1,_)
abstract_claim(student, student,c1,i1)
make_abstract_claim(student, student,c1,i1)
support_abstract_claim(student, student,c1,(s1,s2))
```

(7) SYSTEM_REQUEST

```
["system request STUDENT_EXPLAIN"]
discuss(system,_,_)
claim(system, student,c1,_)
abstract_claim(system, student,c1,i1)
make_abstract_claim(system, student,c1,i1)
support_abstract_claim(system, student,c1,(s1,s2))
```

(8) STUDENT_REQUEST

```
["student request SYTEM_EXPLAIN"]
discuss(student,_,_)
claim(student, system,c1,_)
abstract_claim(student, system,c1,i1)
make_abstract_claim(student, system,c1,i1)
support_abstract_claim(student, system,c1,(s1,s2))
```

(9) SYSTEM_ANNOTATED_EXAMPLE

```
["system critiques own example"]
discuss(system,_,_)
challenge(system,system,c2,i77)
share_claim(system,system,c2,i77)
share_supports(system,system,c2,s1)
new_supports(system,system,c2,i77)
not_share_supports (system,system,c2,s2)
complementary_claim(system,system,c2,i666)
competing_claim(system,system,c2,i999)
```

(10) STUDENT_ANNOTATED_EXAMPLE

```
["student critiques own example"]
discuss(student,_,_)
challenge(student,student,c2,i77)
share_claim(student,student,c2,i77)
share_supports(student,student,c2,s1)
new_supports(student,student,c2,i77)
not_share_supports (student,student,c2,s2)
complementary_claim(student,student,c2,i666)
competing_claim(student,student,c2,i999)
```

```
(11) STUDENT_CRITIQUE
["student critiques system instance"]
discuss(student,_,_)
challenge(student,system,c2,i77)
share_claim(student,system,c2,i77)
share_supports(student,system,c2,s1)
new_supports(student, system,c2,i77)
not_share_supports (student, system,c2,s2)
complementary_claim(student, system,c2,i666)
competing_claim(student, system,c2,i999)
```

```
(12) SYSTEM_CRITIQUE
["system critiques student example"]
discuss(system,_,_)
challenge(system,student,c2,i77)
share_claim(system, student,c2,i77)
share_supports(system, student,c2,s1)
new_supports(system, student,c2,i77)
not_share_supports (system, student,c2,s2)
complementary_claim(system, student,c2,i666)
competing_claim(system, student,c2,i999)
```

We have given these examples in order to show how combinations of dialogue moves relate in general terms to more familiar aspects of teaching and interaction styles in the Intelligent Tutoring Systems literature. The representation is presently rather simple, but is of course extensible in principle. These examples are, however, simply those of interaction styles, and say nothing about aspects of domain traversal, mentioned in the previous chapter. In KANT, this is controlled by dialogue focus⁵ constraints (the spreading activation model), which may be influenced by the new topics introduced by either conversant. In this respect, the interaction styles actually used by KANT will not necessarily correspond exactly to any of the conceptions of teaching styles as currently represented in existing ITSs. It is constructive, however, to consider some examples of how existing styles could be represented using the dialogue moves available in KANT:

Socratic tutoring

At the interaction level, this involves essentially questioning on behalf of the tutor. SCHOLAR (Carbonell 1970) also included a 'Q/A' interaction possibility on behalf of the student. We therefore represent it as combinations of

SYSTEM_ELICIT / STUDENT_ELICIT

[domain traversed by topic 'importance']

Critiquing

This consists of unconnected successions of the user stating a proposed solution which is critiqued:

STUDENT_INFORM, SYSTEM_CRITIQUE

[no explicit domain representation and traversal algorithm]

⁵ The spreading activation model only addresses the problem of defining focus from the point of view of topic-based shifts. As Grosz & Sidner (1986) have recently suggested, the "topic" of discourse may involve a complex interaction between the 'subject' of the discourse and the 'intention' of the speaker.

Drill and practice

The system instructs on some topic, then questions concerning examples, possibly critiquing student solutions:

SYSTEM_INFORM, SYSTEM_ELICIT/ SYSTEM_CRITIQUE

Successive refinement

In Elsom-Cook's (1989a) formulation of this teaching style, a top-down domain teaching strategy is accompanied by general explanations of relevant concepts:

SYSTEM_EXPLAIN

[top-down domain traversal]

Cognitive apprenticeship

Again in Elsom-Cook's formulation, this is a bottom-up teaching style involving annotated examples; it could perhaps be represented by successions of:

SYSTEM_INFORM/ SYSTEM_EXPLAIN/

SYSTEM_ANNOTATED_EXAMPLE/

STUDENT_ELICIT

Clearly, these are not the only possible examples. Within the dialogue move examples given so far, it would be possible for the system to make a claim and to request that the student give possible supports for it. We have therefore quite a large combination of flexible interaction styles which are possible in KANT, and new possible combinations which may not correspond to any of the conceptions of teaching interaction styles which we have presently identified. This is an instance of the view of ITS as being able to suggest new theories of education, which are not necessarily limited by the current educational practices in any specific culture.

Other more general patterns in teaching dialogues have been identified in educational research, which may be characterised by combinations of these dialogue moves, but which one would not necessarily identify as 'teaching styles' as such. An example of this is the 'Initiation-Feedback-Response' cycle, identified by Sinclair & Coulthard (1975) We could represent this simply as combinations of

SYSTEM_ELICIT

STUDENT_INFORM

SYSTEM_CRITIQUE.

In many ways, their analysis of teaching discourse is in accordance with a view of teaching styles as combinations of dialogue moves, since they offer a way of characterising classroom talk in terms of hierarchical categories, from 'lessons' and 'transactions' at the highest level, through 'exchanges', 'moves' and 'acts' at the lowest levels.

In summary, KANT uses a hierarchical tree of 'dialogue moves' as the basic units of educational dialogue. The moves can be viewed as components of existing characterisations of 'teaching styles', and of recurrent structures which have been identified in classroom talk.

5.3.4 Origin of the dialogue moves

In the previous section we described how combinations of dialogue moves relate to existing work on teaching styles. Before moving on to describe dialogue generation algorithms, let us briefly discuss the individual dialogue moves and some relationships with existing work described in Chapter 3.

On the 'CLAIM' side of the dialogue goal tree, the dialogue moves bear close resemblance to the characterisation of 'INFORM' and 'REQUEST' speech acts in the work of Cohen & Perrault (1979). For example, the preconditions for a speaker to 'INFORM' some proposition are that the speaker believes the proposition and that the speaker wants to inform the hearer about this proposition. The effects are simply that the hearer believes that the speaker believes this proposition. In KANT, however, such operators are not formulated into *plans* in the generation of an interactive dialogue.

A number of the dialogue moves are similar to a number of Reichman's (1985) conversational moves. For example, our description of pursuit of goals remaining unpursued after an interruption, would be a 'return' move in terms of her theory, and pursuit of our 'complementary claim' goal would be an 'indirect challenge' move. There are also close affinities between the subgoals to 'challenge' goals which we have specified and approaches to critiquing (Miller 1984) and rules for Socratic dialogue (Collins 1977). There are, however, important differences between these approaches and the way in which certain elements of them have been incorporated into our model as dialogue goals. Critiquing approaches were developed with a view to improving user acceptance of expert systems in medical domains, by providing the user with a critique of their proposed treatment approach, in terms of known risks of drugs and so on, rather than attempting to provide a definitive 'expert' answer. A critique is a single piece of generated text in response to a single proposed solution input by the user. The structure of the interaction is thus a simple answer/response pair, and as such cannot be considered to be a dialogue model in the sense in which we understand it. The model lacks a conception of larger dialogue goal and topic structures, which is hardly surprising since it was never specifically intended to support educational interactions⁶. Similar points could be made about Collins' approach to Socratic dialogues. In his 1977 work, the approach was to specify a set of rules for the tutor's dialogue action in response to certain antecedent conditions present in the student's previous response and certain features of the dialogue state. For example, his Socratic tutoring rule 6 is:

"Pick a counterexample for an insufficient factor.

if

(1) the student gives as an explanation one or more factors that are not sufficient, or

(2) agrees to the general rule in rule 5

then

(3) pick a counterexample that has the right value of the factor(s) given but the wrong value of the dependent variable, and

(4) ask what the value of the dependent variable is for that case, or

⁶ Miller hypothesised that there may be possible educational applications of the model.

(5) ask why the causal dependence does not hold for that case."

In a sense, this is similar to Reichman's (1985) 'direct challenge' conversational move and to the complementary claim goal which we described. Although these rules do embody many certainly true facts (the rules were based on extensive protocols) about tutorial arguments which seek to develop students' critical faculties, we would question the rule-based framework as a formalism for quite low-level dialogue actions. It is possible for a number of tutoring rules to possibly apply at any given point in the interaction, and we are given no educationally based or non-arbitrary means of conflict resolution. In addition, the model lacks any larger scale topic and intentional structures which are macroscopically organised to give an intelligible dialogue structure. In a sense, it could be argued that at a sufficient level of abstraction there is no essential difference between having preconditions for an action which achieves a goal, and having antecedent conditions for executing the right-hand side of a rule. But the essential difference here is that the goal based dialogue move formalism enables us to combine such actions into a sequence of actions which constitute a coherent dialogue, by use of explicit negotiation, in a manner which is not intuitively obvious with chains of rules (Socratic rules are applied singly, with no rational relationship between successive rule firings). Collins' later work (Collins & Stevens 1983) is however, moving more towards a recognition that the tutor's and student's goals need to be accommodated within the tutoring rule system, in terms of decision criteria for selecting cases to be discussed.

In summary, therefore, the principal differences between work on Socratic dialogue, critiquing and Reichman's conversational moves, and the dialogue model which we have discussed are that our model

- integrates intentional and focus based dialogue structures ;
- it has a conception of relevant dialogue focus shifts ;
- it incorporates features which give greater autonomy to the student in terms of symmetrical dialogue structures, and cooperativity through negotiation ; and
- it integrates a number of applicable tutorial styles within a single dialogue model.

5.4 An overview of dialogue generation mechanisms

Now that all fundamental knowledge representations in KANT have been described, we can describe how they are used in dialogue generation. We shall specify the algorithms by expressing the set of Lisp functions which implement KANT in abbreviated English.

5.4.1 The Fundamental Algorithms

5.4.1.1 The ARGUMENT CONTROLLER

At the highest level, the dialogues generated by KANT consist of a succession of *negotiated turns*, which are managed by the ARGUMENT CONTROLLER. A negotiated turn consists of a phase of explicit negotiation, which is followed by the negotiated *dialogue action* only if negotiation succeeds. The dialogue action is not necessarily performed by the conversant who initiates negotiation, hence speaker turns are not necessarily coincident with negotiated turns.

A dialogue action consists of an utterance which communicates some initial belief, which may be followed by further negotiated turns by the speaker who uttered the dialogue action, in order to further explain subconcepts required to understand the initial action. Such negotiated turns may similarly be followed by appropriate dialogue actions, and so on, until negotiation fails, when the initial negotiated turn is over, and the initial dialogue respondent becomes the new initiator of a negotiated turn. Hence, negotiated turns may be recursively embedded within each other, generating a complex dialogue structure, an example of which is illustrated in figure 5.4. Note that this is only one *possible* structure which illustrates negotiated turns, negotiation and dialogue action phases, and recursive structures - other dialogues would generate other combinations of these elements - and the independent structure of topic shifts (although there is a clear relationship between negotiated turns and topic shifts, a change in negotiated turn does not necessarily cause a topic shift).

Figure 5.4
An example high-level dialogue structure of negotiated turns and topic shifts

negotiated turn	n=system s=system	TOPIC
explicit negotiation		t 1
dialogue action		
explicit negotiation		t 2
dialogue action		
explicit negotiation		t 3
dialogue action		
negotiated turn	n=student s=system	
explicit negotiation		
dialogue action		t 4
negotiated turn	n=system	
explicit negotiation		
negotiated turn	n=student s=student	
explicit negotiation		
dialogue action		
explicit negotiation		t 5
dialogue action		

At the highest level, therefore, the dialogue is generated by a simple loop which repeatedly assigns a negotiated turn to each speaker, checking for terminating conditions. Inputs to the

program are

- system belief model (including belief instances assumed to be derived from the musical parser);
- student belief model ;
- the dialogue history ;
- dialogue goal operators ;
- initial parameter values for n, s, c, i .

Since our purpose is to demonstrate the feasibility of a dialogue generation mechanism, we have adopted the following simplifications⁷ in specifying the initial values for these inputs:

- the system is assumed to have a long term memory - represented as a semantic network - which is initialised in terms of node strengths and activation levels as described in an earlier section ;
- the student is assumed to begin the dialogue with none of the beliefs relevant to the future dialogue - i.e. the student belief representation is initially empty ;
- the dialogue history is initially empty ;
- both student and system have access to the same set of dialogue moves ;
- we assume that the system is the first negotiator; concept and instance parameters can be either given an initial value from the system's belief representation, or can be initialised to nil, in which case values will be chosen in the course of dialogue generation (in all future examples, they are initially empty).

We can therefore summarise the simple algorithm for the ARGUMENT CONTROLLER as follows:

function name: ARGUMENT CONTROLLER

arguments:

```
*dialogue_history*
*system_beliefs*
*student_beliefs*
*dialogue_goals*
*parameters* <- ((n system) (s nil) (c nil) (inst nil))
```

function body:

```
loop
  SYSTEM_CONVERSER
  if termination? then RETURN
  STUDENT_CONVERSER
  if termination? then RETURN
```

The predicate 'termination?' returns the value true, leading to return from the dialogue

⁷ We recognise that these assumptions ignore the problem of identifying the student's prior knowledge of various kinds.

generation loop, if either the system or the student wishes to terminate the dialogue. This involves a simple request being generated by the system to that effect, to which the student can answer yes or no. The system informs the student that it is terminating the dialogue when it believes that it can tell the student nothing which the student is not already aware of - i.e. according to the system's model of the student's memory, the student has no memory trace not possessed by the system.

5.4.1.2 The CONVERSER

The CONVERSER is divided into two sections - one for each speaker - involving functions which have only minor differences, in terms of input and output generated. We shall therefore describe the SYSTEM CONVERSER, then describe how the STUDENT CONVERSER differs from this.

At the most general level, the CONVERSER uses an algorithm for searching the goal operators, beginning with the highest level goal 'DISCUSS'. It selects the appropriate variant of the CONVERSER for each negotiator (system or student), and attempts to find a single path of goals whose preconditions can be satisfied, given the same inputs as the ARGUMENT CONTROLLER. Examples of such goal sequences were given earlier (§5.3.3). The function which controls this search is called `search_goal_tree`. Its operation is very simple:

- search the goal tree, beginning at the highest level goal DISCUSS (the initial 'current goal'); stop if there are no parameters or there is no current goal ;
- *converse* the current goal, returning updated parameter values for successful conversing, or no parameters if conversing fails ;
- from the subgoals of the current goal, choose a new goal as the current goal if the goals are 'exclusive-or', or search the subgoal tree under each subgoal otherwise.

This may be summarised as follows:

function: `search_goal_tree`

arguments:

`dialogue_history`

`system_beliefs`

`student_beliefs`

`dialogue_goals`

'((n system) (s nil) (c nil) (inst nil)) # | parameters | #

'DISCUSS # | current goal | #

function body:

terminate if there is no current goal

terminate if parameters are nil

`converse_a_goal`

converse the current chosen goal # | initially this is 'DISCUSS' | #

if `converse_a_goal` succeeds, return updated parameter values with which it succeeded, return nil parameter values otherwise

discriminate_subgoals

select a new current goal to converse, from the exclusive-or subgoals of the current goal

search_goal_tree

with new current goal and parameter values inherited
from parent goal, or for each of a list of conjunctive subgoals

5.4.1.3 Discriminating between alternative dialogue subgoals

When subgoals are 'AND-OR', we recursively apply the function for searching the goal tree on each of the subgoals, depth-first, left-to-right. In the case of 'EXCLUSIVE-OR' subgoals - for example in choosing whether to CLAIM or CHALLENGE - a mechanism for discriminating between them to choose the next current goal is applied (the function *discriminate_subgoals*). Discriminating subgoals means choosing one to attempt to converse *under specific parameter bindings* - i.e. to negotiate and to try to perform any dialogue action associated with the subgoal. Initially, this involves evaluating the dialogue state preconditions of each goal, using the EVALUATOR, to determine whether a goal is possible in the current dialogue state under given parameter bindings. If only one subgoal can be satisfied in this way, then this is the new current goal. In the case of exclusive-or subgoals for CLAIM and CHALLENGE, for example, it will often be the case that only the CLAIM goal can be satisfied, when no previous claim requiring critique (CHALLENGE) is present in the dialogue history. When the preconditions for challenging can be met, however, it is also very often the case, that we could also choose to make a fresh claim instead. In cases such as these, therefore, we must *discriminate*. Discrimination is done by using a set of *educational principles*, derived from some aspects of current research on educational discourse. With our present set of dialogue moves, only two principles are required (others may well be identifiable from educational research):

- (1) Prefer elicitation⁸ (try to satisfy student as speaker).
- (2) Prefer to challenge a previous ('challengable') claim rather than make a new claim.

The origin of these principles in educational research is as follows.

Principle (1) reflects the fact that a negotiator must choose for a given dialogue move, whether they wish to be the speaker, or whether they wish the respondent to be the speaker. For example, the system must choose for a CLAIM goal whether they or the student should try to make that claim. We use the principle of trying to elicit the student as speaker as much as possible, i.e. elicitation rather than instruction.

Principle (2) reflects the existence of 'INITIATION-RESPONSE-FEEDBACK' cycles in teaching dialogues, as identified by Sinclair & Coulthard (1975), corresponding to a succession

⁸ This is derived from Bellack & others' (1966) empirical studies of the 'teacher's moves' in classroom discourse.

of CLAIM/CHALLENGE moves. At present these principles are embedded in the program, but we recognise that for a larger set of dialogue moves a set of such principles would need to be identified and given an independent declarative representation.

The procedure takes each subgoal in turn, and for each attempts first to satisfy its preconditions with the student as speaker, then with the system as speaker if this fails. If both can be satisfied with the student as speaker, then challenging is preferred over claiming. If only one is satisfiable with the student as speaker, then this is chosen. The function returns the chosen subgoal, and updates the parameter values under which this goal is satisfiable as input to the function for conversing that goal. We can therefore summarise the discrimination procedure as follows:

function: discriminate_subgoals

arguments:

subgoal_list

dialogue_history

system_beliefs

student_beliefs

parameters

function body:

for each subgoal

if

evaluate_ds_preconds satisfied with s<-student substituted into dialogue state
preconditions of subgoal,

then

put paired subgoal & satisfiable parameters into
subgoal_discrimination_list

if

evaluate_ds_preconds satisfied with s<- system,

then

put paired subgoal & satisfiable parameters into
subgoal_discrimination_list

if

only one member in subgoal_discrimination_list,

then

return this subgoal as new goal, and pass satisfiable parameters to goal
converser

if

more than one goal in subgoal_discrimination_list,

then

return the subgoal as new goal, and pass satisfiable parameters go goal
converser which has student as speaker, or if both have student as speaker,
the subgoal which involves challenging rather than claiming

if

no subgoals satisfiable, return nil satisfiable parameters and nil new goal

<causes converser to terminate>

The function `evaluate_ds_preconds` takes a set of input parameter values, substitutes their values into a dialogue state precondition expression in a goal, and evaluates the substituted expression with respect to the current dialogue state (a set of global variables in the current Lisp environment). As described earlier, evaluating the expression involves calling a number of functions which access global variables for the participants' belief sets and the dialogue history, so that the composite expression returns 'true' (i.e. non-nil) if it is satisfiable. We shall discuss the details of these individual preconditions in a later section.

5.4.1.4 Conversing a single dialogue goal

The conversing of each individual dialogue goal operator is responsible for the generation of nearly all textual communication between the two conversants. Exceptions are the theoretically uninteresting pieces of text which state that a negotiated turn is over, and that the conversants can agree to terminate the dialogue. In simple terms, a goal is conversed by

- negotiating the parameter values under which its dialogue state preconditions can be satisfied;
- performing dialogue actions with substituted negotiated parameter values ; and
- propagating effects of negotiation on the dialogue history and of the dialogue action on the previous dialogue state.

The major components are therefore:

EVALUATOR

- substitutes parameter bindings into dialogue state preconditions and attempts to satisfy them with respect to the current dialogue state;

NEGOTIATOR

- negotiates these bindings

INFORMER

- gives text explanations using the COMMUNICATOR, and recursively invokes the CONVERSER if necessary, using the COMMUNICATOR to generate appropriate presequences.

EFFECTOR

- propagates effects of INFORMER on the dialogue state - belief sets and dialogue history.

The EVALUATOR consists of a component which substitutes input parameter values into a dialogue state precondition expression, and a component which evaluates that substituted expression. In the case of exclusive-or subgoals, the EVALUATOR is invoked as part of the process of discriminating between subgoals. In the case of conjunctive subgoals, the EVALUATOR is invoked as part of the discrimination process. In all cases, the value returned is the set of parameter bindings under which the dialogue state preconditions can be satisfied. As with the discrimination process, the program initially tries to satisfy the preconditions with the

student as speaker, before trying with the system as speaker, if at this point in the goal hierarchy, the inherited parameter for 'speaker' is not already bound. As part of the evaluation process, new parameters may be bound if the function 'bind' is encountered within the precondition expression. For example, if we enter the CONVERSE_A_GOAL function with the bindings

```
((n 'system) (s 'student) (c 'harmony) (inst nil))
```

[i.e. 'the system negotiates that the student should be the next speaker concerning the concept 'harmony']

for the goal CONCRETE_CLAIM, then a function within the precondition expression will try to find a value for a concept to talk about, in terms of a complex algorithm which integrates focus and learning constraints (to be discussed in section 5.4.2):

```
(bind 'inst
      (select_instance_to_talk_about
        (get_unmentioned_instances
          'harmony
          *dialogue_history*
          (get_instances 'harmony (ltm_model 'student)))))
```

If a (non-nil) value can be found, then it is returned and thus this part of the expression will evaluate to 'true'. In this example, if the student knows no instances of 'harmony', then the system will not ask for the student to state one. Any new value found in this way is returned as part of the final set of parameter bindings returned by the EVALUATOR - in our example above, if satisfied, the following parameters might be returned:

```
((n 'system) (s 'student) (c 'harmony) (inst 'i987))
```

If the EVALUATOR returns some non-nil value for satisfiable parameters, then these are passed to the NEGOTIATOR. Negotiation involves the negotiator explicitly proposing the parameter bindings which are input to the NEGOTIATOR to the respondent, in the form of simple text templates. We shall discuss the negotiation process in more detail in section 5.5.2, but for the moment we are simply interested in the functional role of the NEGOTIATOR in the process of conversing a single goal. Supposing we negotiate the example parameter bindings above:

```
((n 'system) (s 'student) (c 'harmony) (inst 'i987))
```

then the system would generate text of the form

"Do you accept that <student> makes a <concrete_claim> about instance <i987> of concept <harmony>?"

If as a result of negotiation, the respondent accepts, then the goal succeeds with the negotiated parameters, which are then passed to the INFORMER and the EFFECTOR.

The INFORMER has the function of communicating beliefs. It consists of an expression into which the negotiated parameters are substituted, which is then evaluated. For example, the form

```
(INFORM s c)
```

is substituted to give

```
(INFORM 'system ('chord))
```

which is then evaluated, resulting in the generation of the text associated with the system's semantic memory node 'chord'. The expression can take varied arguments, depending on whether a general concept, an instance, a justification type or instance is to be communicated. Each of these possible argument forms has a different associated text template, and a different procedural interpretation - in some cases the INFORMER recursively reinvokes the CONVERSER (described in section 5.4.1.2), and in others it does not. We shall deal with these details in the later section 5.5.3.

When a dialogue participant makes some statement, this may have effects on the mental states of both participants. After a dialogue action has been performed, therefore, the EFFECTOR is invoked. This evaluates the substituted expressions contained in the 'negotiation_effects' and 'action_effects' components of the goal operator currently being conversed. Negotiation effects simply record the agreed parameter values and the dialogue move for which they were agreed in the 'dialogue history' (described earlier). An example substituted negotiation effects expression is:

```
(negotiation_effects=
  ((update_dialogue_history '((goal_name= 'CHALLENGE)
                              (c= 'phrase_boundary)
                              (s= 'student)
                              (inst= 'p6))))))
```

The substituted expression - "the student made a challenge concerning instance p6 of concept phrase_boundary" - is simply inserted at the beginning of the dialogue history.

Action effects are somewhat more complicated. They involve updating the participants' belief sets in the way described in the earlier section §5.1.2.2⁹. For example, the action effects expression invoked when the system was the speaker for the corresponding dialogue action is:

⁹ In brief, any dialogue participant who does not have a node for that concept in their memory representation is assumed to acquire one, the concept becomes an activation source, and the node (memory trace) strength is incremented.

```
(action_effects=
  ((update_ltm_models '((c= 'phrase_boundary) (inst= 'p666))))))
```

This would have the effects that

- 'phrase_boundary' becomes an input activation source in the memory models for both participants ;
- if the respondent does not possess a concept node for this concept and/or a record of this instance, then this is created in the initiator's model of the respondent ;
- the node strength for this concept is increased in the memory representation for each participant.

Let us now summarise the general control structure for conversing a single goal:

function: converse_a_goal

arguments:

dialogue_history

system_beliefs

student_beliefs

dialogue_goals

'((n system) (s nil) (c nil) (inst nil)) #| parameters| #

'CLAIM #| current goal| #

function body:

evaluate_ds_preconds

substitute input parameters into dialogue state precondition expression for current goal; attempt to satisfy with student as speaker, otherwise with system as speaker; return any new bindings created during evaluation of preferred speaker value, and update parameter values with these; return nil if cannot be satisfied

negotiate parameters provided non-nil; return agreed parameter values

inform agreed parameter values, provided non-nil & there is an action to perform

propagate_effects

substitute informed parameter values (if an action has been performed); evaluate action_effects & negotiation effects.

5.4.1.5 Variants of the dialogue generation programs with the student as negotiator

In this section we shall briefly describe the differences in the programs used for generating the system's and the student's negotiated turns. One of the major design goals of KANT was to generate dialogues which are as *symmetrical as possible*, in terms of giving the student as much access as possible to the interaction styles available to the system, and in the kinds of representations used for conversants' beliefs. The differences between the generation programs for the negotiated turns of the system and of the student are, therefore, relatively minor, and largely concern direction of printed input and output to the program.

Student conversing

On a student negotiated turn, the mechanisms involving searching the goal tree are essentially the same. At the position where the system has to choose a goal or goals from the available subgoals, we assume that the student has some (possibly unconscious) set of criteria for deciding which goal to pursue next. The system therefore searches the goal tree and presents the available choices at each point, allowing the student to choose. Where subgoals are exclusive-or - such as between CLAIM and CHALLENGE - the student is invited to choose a goal by the generation of simple text templates:

"system:

Would you like to choose a goal from the alternatives

CLAIM (a statement about some topic)

CHALLENGE (challenge a previous claim)

student: *claim*"

The student is then invited to choose the parameters for that goal; for example:

"system: who do you want to be the next speaker? (system or student)?

student: *system*

system: what topic do you want us to discuss? (type 'show' to see possibilities)

student: *key*

system: type in the specific instance of this concept that you want us to discuss (if any)

student: *nil*"

Negotiation

The system now records that the student has chosen to negotiate that the goal CLAIM is to be discussed, with the following parameters (the 'student negotiated parameters'):

((n student) (s system) (c key) (inst nil))

The system must now decide whether to cooperate with this proposal or not. In the present implementation, we have been concerned simply to place an explicit negotiation phase within a dialogue generation mechanism, and have not extensively explored the complex structure of such metadialogue utterances. A number of initial simplifications have therefore been made. In this case, the system adopts the assumption that it will decide whether to cooperate by 'putting itself in the student's position', and attempting to satisfy the preconditions for pursuit of that dialogue move, from its own point of view. It therefore simply invokes the EVALUATOR, described earlier, with the student proposed parameters as input: if the dialogue state preconditions for the current goal chosen by the student can be satisfied with respect to the current dialogue state, the system cooperates, otherwise it does not. It generates simple templates as follows:

Either

"I agree to cooperate with your proposal that <speaker> pursues the goal of <goal> about the concept <concept>"

or

"I do not agree to cooperate with your proposal that <speaker> pursues the goal of <goal> about the concept <concept>"

Dialogue actions and effects

If the system cooperates, then any dialogue actions are performed, and negotiation and action effects are propagated in an identical manner as for system negotiated turns. In the case of actions where the student is a speaker, the system simply generates a summary statement of the student's choices, which the student is asked to confirm. If the system has been chosen as speaker, then it generates INFORM actions in an identical manner, as if it had itself chosen to be the speaker on its own negotiated turn. For example, suppose the student has chosen to pursue an ABSTRACT_CLAIM with the parameters as above, and the system has agreed to cooperate:

```
((n student) (s system) (c key) (inst nil))
```

In this case, the appropriate action is

```
(INFORM 'system ('key))
```

which results in the system generating explanatory text for this concept, and may result in the system subsequently taking a negotiated turn which results from a recursive application of the CONVERSER on subconcepts of 'key' concerning which it believes the student has insufficient knowledge to understand the initial explanation. Supposing that the student wished to make a CONCRETE_CLAIM, then the chosen parameters could be

```
((n student) (s student) (c key) (inst 'p99))
```

The INFORM statement now takes the form

```
(INFORM 'student ('key 'p99))
```

When the first argument to the INFORM statement is 'student', then a summary text statement of the following form is generated:

"system: You have made a <goal name> that <instance> is an instance of the concept <concept>. Can you confirm this (y/n)?

student: y

system: Ok, I will try to remember that"

In the case where the student does not confirm, then no action effects are propagated, and the

student is given the option of a further negotiated turn. Negotiation effects - in terms of recording the goal name and its parameters - are propagated in an identical manner.

There is one difference between system and student negotiated turns which is *not* trivial, and which forms an important area for further research. This concerns the fact that

the possibility of making abstract claims, or general explanations of concepts in natural language, is not available to the student.

This is because the system has no sophisticated mechanism for understanding unrestricted natural language input¹⁰ on the part of the student. Even if we had such a capability, at our present state of knowledge of mechanisms for argument acceptance and psychologically plausible belief revision, it is not clear how the system could respond in an intelligent way. Such questions would form the basis of other complete research programs, which can be considered to be reasonably beyond the scope of the present research described here, which focusses on general mechanisms for generation of tutorial dialogues.

The means by which the student is able to pursue his/her own learning goals in the interaction will be made more apparent in discussion of sections of dialogue generated by KANT in section 5.6

5.5 Some components of the dialogue model in more detail

Not all aspects of the design of KANT were fully implemented in the first prototype. In the following sections, we shall therefore describe the actual implementation, together with proposed extensions towards the 'ideal' model of Negotiated Tutoring described in the previous chapter. These extensions constitute future research.

5.5.1 Dialogue State Preconditions

It is clear that at any point in a dialogue, there is a limited space of utterances which are appropriate (Grice 1975). For example, a speaker cannot ('reasonably') dispute a claim if no claim has been made to dispute¹¹. Within this space, some kinds of utterances seem to be preferred in 'cooperative dialogues'. For example, it seems reasonable that each speaker should try to achieve a balance between relating their utterances to those which have already been made, and making fresh statements about new topics (Grice 1975). The definition of this space of utterance types which are appropriate at a given point in the dialogue is achieved in KANT by representation of *dialogue state preconditions*. They are 'preconditions' in the sense that they express the very general minimum conditions for the appropriateness of a particular dialogue move at a point in the dialogue. As discussed in the previous chapter, we do not believe that such intuitions are captured by the notions of grammaticality or well-formedness, since human dialogues appear to be sufficiently flexible that almost any form of utterance may

¹⁰ Nor does the system possess sophisticated 'knowledge acquisition' tools, as discussed in further work (chapter 6).

¹¹ We exclude the case where a speaker does not 'state' a claim, but 'mentions' a possible hypothetical claim which *could* be made: mentioning is here viewed as stating.

coherently follow any other, given certain contexts. Levinson (1983) has argued persuasively that these notions of coherence in dialogue are better captured by the concept of the *expectations* of speakers. In KANT the succession of dialogue moves is in fact controlled by *three* interrelating factors:

- (1) *dialogue state preconditions*,
which delineate a restricted space of applicable dialogue moves with respect to the current dialogue state;
- (2) *educational principles*,
which may be considered as very high level general educational *goals* for choosing a preferred move from the space defined by (1);
- (3) *negotiation mechanisms*,
which operate to attempt to secure cooperativity for the move chosen by (2).

Focus constraints, are not really considered as a separate category here: they are represented as part of the dialogue state preconditions which constrain applicability of those moves which involve possible focus shifts. The notion of the 'dialogue state' was discussed in the previous section 5.1.1.

Preconditions are not, however, simply *passive* in their nomination of possible goals: if a speaker attempts to nominate a particular type of dialogue move - such as a concrete claim - then procedures attached to precondition expressions will sometimes *actively* search for parameter values under which those precondition expressions could be satisfied, such as in the identification of a new concept or instance value which could enable a precondition expression to be satisfied.

5.5.1.1 Examples of dialogue state preconditions

Let us now discuss details of a number of these preconditions. We shall describe them beginning with a general statement concerning the preconditions for each goal, and proceeding to give details for certain important recurring preconditions.

At the most general level, the goal is that a 'discussion' should take place:

DISCUSS

satisfied if for any speaker, provided its representation of its own and the respondent's beliefs are not identical

Preconditions for CLAIMing

The following preconditions are for dialogue moves on the 'CLAIM' branch of the goal tree.

CLAIM

satisfied if

either

**a concept has already been chosen about which a claim is to be made,
and nothing has yet been said in the dialogue,
and the concept is in focus for the speaker
and the respondent does not already understand that concept
and the speaker has a memory trace for that concept**

or

**if no concept has already been chosen to discuss
and some concept can be chosen which meets criteria concerning the extent to which a
concept is 'understood' according to the memory models of each speaker**

We shall discuss what it means for a concept to be in focus for a speaker, and how new concepts are chosen, later in this section. If nothing has yet been said, it seems most reasonable for a speaker to choose that something should be said about some concept.

CONCRETE_CLAIM

satisfied if

either

**there is some concrete instance to discuss
and the instance is not known to the respondent
and the speaker has a memory representation of that instance**

or

**some instance can be chosen which meets certain criteria in terms of the memory models of
both participants**

We shall discuss how a concept instance may be chosen in a later section.

MAKE_INSTANCE_CLAIM

satisfied if

there is a concrete instance which has been inherited, about which to make a claim

SUPPORT_INSTANCE_CLAIM

satisfied if

**an instance has been stated for which supports can be given
and the instance has supports in the memory representation of the speaker
and the concept of which an instance has been stated is known to both participants
and the respondent does not know these supports**

ABSTRACT_CLAIM

satisfied if

either

for the concept to be discussed, no instances exist which have not been stated by the speaker

or

the previous goal was a concrete claim

The preconditions for making an abstract claim - i.e. a general explanation about some concept - are reasonably restrictive, in order to give us a reasonably clear mechanism for deciding whether a concrete or an abstract claim should be pursued. We have adopted the general approach of allowing speakers to move from *specific to general* in making successive claims, and that if we can possibly state something concrete, then this should be preferred initially over generality. In very general terms this would correspond to a general strategy of *concretising* discussion of concepts with initial examples.

MAKE_ABSTRACT_CLAIM

satisfied if

the system is the speaker

and the system has a memory trace for that concept

and the student does not have that memory trace

and the 'instance' parameter has presently no inherited value

The preconditions of this goal reflect the limitation mentioned above that only the *system* has the possibility of making abstract claims, being a serious limitation in the dialogue symmetry provided in KANT. The final precondition here checks that the 'instance' parameter has not already been bound at some higher point in traversing the hierarchy of goal operators, in which case a concrete claim could possibly be made. This could be the case, for example, if the system was initialised by a user who wanted to specify the initial instance which should be discussed, as well as the specific concept.

SUPPORT_ABSTRACT_CLAIM

satisfied if

some concept has already been discussed by an abstract claim

and some set of general support types for this concept can be found which the respondent does not know

Supporting abstract claims about concepts involves stating general explanations about the kinds of justifying evidence which would be relevant to supporting claims about that concept. To consider the hypothetical geological dialogue described in the previous chapter, for example, part of our understanding of the concept of the mineral 'calcite' involves understanding that the justification types of 'crystal form' and 'chemical composition' would be relevant to understanding 'calcite'. The relevant *justification instances* here - given as 'SUPPORT_CONCRETE_CLAIM' subgoals - would be some specific crystal form, such as 'cubic' and some specific chemical composition, such as 'CaCO₃'.

Generality and specificity of preconditions

Within the preconditions discussed for CLAIM moves, we can see that the relationship between generality of moves for CLAIMing in general, to increased specificity in the making and supporting of concrete and abstract claims, is accompanied by a parallel move from generality to specificity of dialogue state preconditions. This is reasonable if we consider the difference in a conversant's reasons for acceptance of 'the making of some statement about a concept' and reasons for agreeing that 'the student should state a specific instance of some concept'. The more general a goal is, the less stringent will be its conditions for applicability. Specificity is generally accompanied by binding of unbound parameters - for instance, it does not make sense to bind an 'instance' parameter before we have decided whether to make a concrete claim or an abstract one.

Preconditions for CHALLENGES

Preconditions for challenges refer more specifically to the dialogue history in terms of previous dialogue moves, whereas CLAIM preconditions clearly need to refer more precisely to participants' beliefs and to focus constraints on the proposed new topic of the claim. We do need, however, to check that the new material introduced as part of a challenge is not already known - a feature which relates to Grice's (1975) 'maxim of informativeness', part of which states "do not make your contribution more informative than is required".

CHALLENGE**satisfied if**

within the last negotiated turn, according to the dialogue history, the move pursued was the making or supporting of an instance claim,
and, the proposed speaker has a memory trace corresponding to the concept for the previous instance claim to be challenged

At this level, the preconditions are very general:

- there must be a previous specific claim to be challenged, and this must be a *concrete_claim*. Since the subgoals for abstract claims involve the making of general claims, expressed in generated text, there is no possibility for either the system or the student to challenge the validity of these general claims in the present implementation. This relates to the asymmetry described earlier, in terms of the inability for the student to make abstract claims. In the present implementation of KANT, we have adopted the simplification that the claim challenged must have been made in the previous negotiated turn. In future implementations, this restriction will be relaxed, and a new precondition inserted, as for CLAIM goals, to check that the concept introduced in the challenge remains sufficiently in focus¹².
- before a speaker can challenge a previous claim about a concept, they must themselves have some beliefs - a memory representation - concerning that concept with which to challenge it (!)

¹² There are similarities between such challenges of previous goals, and the resumption of a previous focus, with Reichman's (1985) 'return' conversational move.

We do not at this stage require that a challenger have something new to say in terms of this challenge: they may simply state that they share or do not share the previous belief stated, even though they may have nothing new to say in terms of new supports and so on.

CLAIM_SHARED

satisfied if

the previous instance stated in a concrete claim is present in the speaker's belief set for that concept

CLAIM_NOT_SHARED

satisfied if

instance challenged not a member of belief set of challenger

In the present implementation, a speaker may not challenge their own claims. Future implementations of KANT could attempt to include this capability with the following modifications:

- Insert a new precondition in CHALLENGE goal which removes the restriction on challenging to previous negotiated turn.
- Insert new precondition to CHALLENGE goal which checks that the concept of the new challenge is *sufficiently* in focus, as for CLAIM goal. We need not require that the new focus is necessarily the highest activated concept, since there is a difference between restrictions in dialogue on the introduction of completely new foci, and the resumption of old foci. If, according to some measure, the old focus is still sufficiently active, then it may be resumed. It is not clear at this stage of research how we could specify such a measure of 'focus distance' other than relying on activation levels.
- Insert a new subgoal to CHALLENGE goal, called RESUME_CHALLENGE, with identical subgoals to AGREE_CLAIM and DISAGREE_CLAIM: in the case of a conversant returning to challenge their own previous claim, they will clearly 'agree' with it, in a sense, since they have themselves stated it. This will be true until future versions of KANT have developed mechanisms for 'argument acceptance', discussed in the previous and future chapters. In this case, it would be possible for the system, for example, to alter its belief set in response to student utterances, and then return to challenge its own previous claim. We presently require no restriction on the student returning to disagree with their previous claims, but would require an extension of the mechanism for updating beliefs, which removed a belief from the system's representation of the student's beliefs in such an eventuality, and maintains consistency of belief sets.

The subgoals for disagreeing and agreeing with claims are identical, but differ in their ordering, as discussed earlier (section 5.3.2.2). This is because when disagreeing it is most natural to begin with those factors with which one disagrees, and the converse with agreeing with claims (Weiner 1980).

COMPLEMENTARY_CLAIM

satisfied if

the previous instance claim was accompanied by a set of justifications, and some instance can be found in the speaker's belief set which is not identical with the challenged instance and which possesses either an identical set to the justifications of the challenged instance, or which contains a superset of those justifications

Statements about goals to be pursued are accompanied with summary statements about what those goals mean - for example

"COMPLEMENTARY_CLAIM

(= an instance which has same set of justification types as previous challenged claim)"

Therefore, when a new instance is stated as part of the action component of this goal, it will be appropriately understood. The purpose of this goal is to give the respondent some reason for considering the complementary claim as a possible instance, in terms of their previous statement. Again, in terms of the simplification in the present implementation stated earlier, speakers will not presently make complementary claims with respect to their own previous claims. We could, however, envisage a situation in dialogue when it might be appropriate for a speaker to say something of the form:

"You remember previously that I said that p1 was an instance of a phrase boundary, because of the chord progression present at that point? Well, I've just realised that the progression is also present at position p1001 in the melody, so I think that it's also a phrase boundary, for the same reason..."

The following goals all involve stating something about the *supports* stated for a previous claim¹³. Their preconditions are all therefore quite straightforward, and have a lot in common with each other, given that more general preconditions for CHALLENGEing have already been established at this lower point in the goal hierarchy.

AGREE_SUPPORTS

satisfied if

the challenged instance was given some non-nil set of justifications

and if there is a set of justifications which exist as a subset of the justifications for that identical belief instance in the speaker's memory representation for the appropriate concept node

¹³ An important area of Artificial Intelligence research is concerned with 'truth maintenance systems' (Doyle 1979). This is a method for logical reasoning which uses the notion of 'supports' for beliefs derived in the course of a theorem proving process. We did not attempt to address this research in our present work, since (1) we do not include an explicit logical reasoning mechanism in the system; (2) our research was concerned with generating reasoning *in dialogue* (or 'argument'), (3) truth maintenance systems do not claim to model *human* reasoning, (4) we wanted our dialogue model to be based on a *cognitive* theory of memory (Anderson 1983). The combination of truth maintenance systems with psychological memory and reasoning constraints remains a possibility for long-term research.

DISAGREE_SUPPORTS

satisfied if

the challenged instance was given some non-nil set of justifications
and if there is an set of justifications in the justification stated for the challenged belief
which are not a subset of the justifications for that identical belief instance in the speaker's
memory representation for the appropriate concept node

NEW_SUPPORTS

satisfied if

there is an set of justifications in the justification set for the concept node in the speaker's
memory which matches the challenged claim, which are not a subset of the justifications
stated in the challenged claim

The purpose of these dialogue moves is *not* necessarily to instruct the student about what is or is not 'certainly' the case: since the system is itself not certain that its beliefs are certain. In terms of the plausible reasoning model described in the previous section to this thesis, it simply communicates the way in which its own beliefs relate to those of the student, thus giving an alternative point of view which may allow the student to rationally revise his/her beliefs.

5.5.1.2 Preconditions for determining focus, concepts and instances

We have now discussed complete sets of dialogue state preconditions for example dialogue moves, and move on to describe a number of specific preconditions which recur throughout a number of moves, for checking whether proposed concepts are sufficiently in focus, whether possible concepts and instances should be discussed, and choosing new concepts and instances to discuss where required.

Deciding if a concept is in focus

In discussing dialogue moves for CLAIMing, preconditions were mentioned which check whether a proposed new concept to discuss is sufficiently in focus or not. When a concept is explicitly mentioned in dialogue, a copy of its long-term memory representation is created in working memory, along with concepts to which it is closely connected, in virtue of their activation. In terms of the Andersonian theory which we are applying here, membership of working memory is a matter of degree. We can therefore view the newly activated concept as the current *explicit dialogue focus*, and the other members of working memory as potential *implicit foci*. In a sense, therefore, implicit focus is also a matter of degree, and there must be a continuum between concepts which are possible foci and those which are not in terms of continuous variations in activation levels. For the purposes of defining a decision mechanism concerning constraints on focus shifts in the dialogue, we have therefore had to make a number of decisions concerning the relative activation level required for a concept to be 'sufficiently in focus'. These decisions are therefore largely *arbitrary* in a numerical sense, but non-arbitrary in the sense that they are based on a psychological theory of memory: we simply claim that some such decision is made in dialogue, but make no psychological claims concerning the precise numerical level used, given that some level must be chosen by a computer program which aims

to simulate these decisions. Psychological theories of memory have long established that working memory has a limited capacity, and that this capacity is approximately *seven 'cognitive units'*¹⁴ (Simon 1982). We have therefore adopted the policy of viewing the seven concepts with the highest activation level as the set of *possible new foci*. We therefore have quite a simple mechanism for establishing whether a proposed new concept focus is a possible new focus which will be within certain constraints:

For a given speaker and a given concept, the function 'in_focus?' retrieves the belief representation for that speaker, and finds the seven concepts in that representation which have the highest current activation level (excluding cases where concepts have nil activation). *If the proposed new focus is a member of those concepts which are viewed as the current working memory, then it is a possible new focus, otherwise it is not.*

We shall discuss the implications of this view of dialogue focus in comparison with existing theories in the next chapter.

Choosing a new concept to discuss

A reasonably complex algorithm is involved in the mechanism by which the system chooses a new concept to discuss (such as in the CLAIM goal). The algorithm tries to achieve a balance between the following factors:

- (1) choosing the concept with the highest activation ;
- (2) choosing a concept which has fewest subconcepts which are unknown to the student ;
- (3) choosing a concept which is 'important'.

(1) choosing an active concept

As with our discussion of deciding whether a concept is in focus or not, the set of concepts which are sufficiently active is the set which are in the current working memory. The first part of the algorithm for choosing a new focus therefore involves determining the current working memory, *i.e. the seven concepts with the highest activation level, and determining the highest.*¹⁵

(2) choosing a concept as a learning goal

We have adopted the view that since the explanation of a new concept involves explanation of its subconcepts, it is best to begin by discussing those concepts which have fewest unknown subconcepts. This is determined by following the subconcept links from each of the seven concepts in working memory, and retrieving the complete subconcept network beneath each,

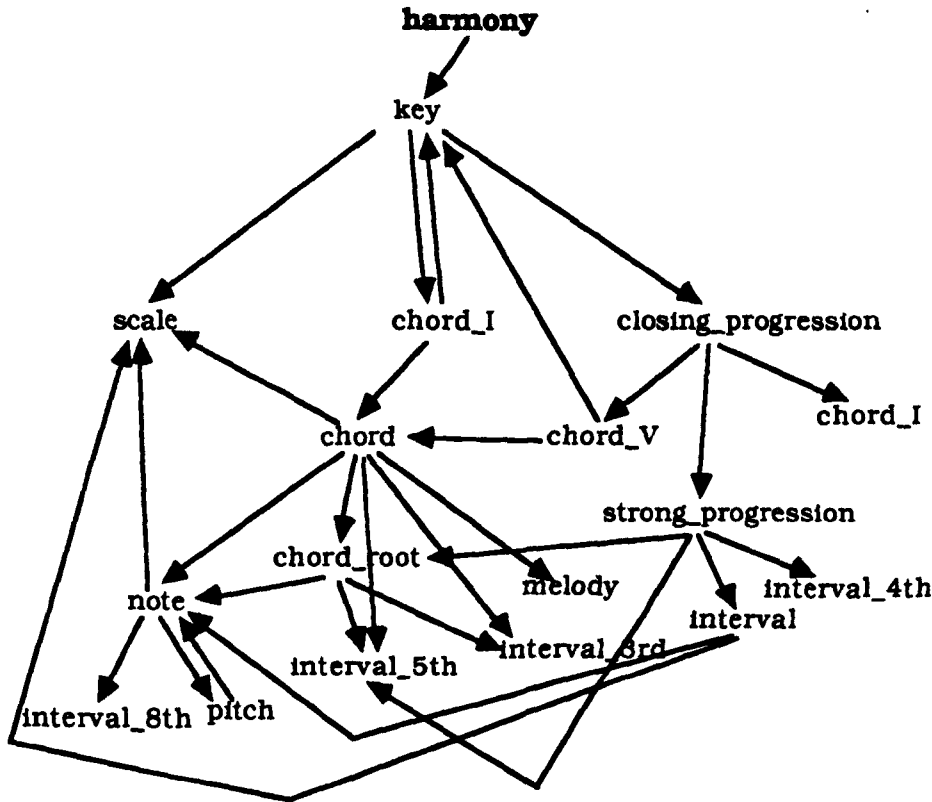
¹⁴ We recognise that this statement depends crucially on the way in which cognitive units are 'chunked'. In any event, the truth or falsity of this theory is not at all crucial to our theory of dialogue. Since 'relevance' must be a matter of degree, in terms of the theory of memory which we have applied here (Anderson 1983), this conception of working memory simply provides an implementationally convenient data representation, which avoids our having to search the entire network on each of the multiple occasions during dialogue generation that we need to check focus constraints.

¹⁵ We again emphasise that a separation of working memory in this way merely provides a convenient independent data representation, which avoids repeated checking of the network: working memory in this sense is simply a highly active area of long-term semantic memory, combined with a short-term memory.

which would need to be taught as part of that concept. Figure 5.5 is an example of such a subconcept network beneath the concept 'harmony'.

Figure 5.5

Example subconcept network, beneath the concept 'harmony'



We can see that the network is in fact a *directed graph*. However, since its directedness is not recorded *explicitly* in the network - it is implicit in the search algorithms which operate on the network - the algorithm for retrieving such trees keeps a record of which nodes have been passed through, to avoid *some* repeated cycling through the network¹⁶. Whether a concept is 'known' or not, is a matter of degree - just as whether a concept is part of working memory or not is a matter of degree - and relates to the *node strength* of a concept. Node strength is a function of how often the concept has been recently mentioned, as described in the earlier section on Anderson's semantic memory theory. Therefore, we can say of a subconcept tree in the system's memory, that it is more 'known' to the student the higher is the sum of the node strengths of the corresponding concept representations in the student's memory¹⁷. The second part of the

¹⁶This can not provide a *general* method for avoiding cycling in searching graphs. Our purpose in searching the network here is simply to derive a list of different subconcepts attached to a concept, in order to produce a sum of their activation level. We emphasize that we are not using the semantic network for reasoning.

¹⁷This is merely a convenient approximate numerical measure, which is based on the encoding

concept choice algorithm is therefore to take each of the subconcept trees which have been isolated as the system's working memory, and assign a numerical score to each, which is equal to the sum of node strengths of each of the corresponding nodes in the student's memory. Clearly, if there is no corresponding node for any concept in the representation of the student's memory representation then no value is included in the sum (i.e. the student is assumed to be 'unaware' of that concept). We can now isolate a subset of working memory which represents the minimal extension of the student's belief representation. If there is a single concept which satisfies the condition of having the highest 'known-ness' score, then this is chosen as the new concept. If there is more than one which has equally the highest score, then the most 'important' of these is chosen¹⁸.

(3) *choosing an important concept*

If we consider the concepts in any domain, there are usually some which are 'central', in the sense that they would need to be understood as part of the explanation of several concepts in the domain. In terms of a semantic network representation which relates to 'associations' between concepts, an 'important' concept will therefore be one which is a subconcept of several concepts and which points to several concepts. One concept is more important in our sense than another if it has a greater number of incoming and outgoing concept links¹⁹. For example, in the example above (figure 5.4), the concept 'note' is more important than the concept 'chord_I'. In terms of the general algorithm for choosing a new concept, if we have several concepts which are suitable learning goals, and which are sufficiently active, then the most important is chosen, in terms of the sum of incoming and outgoing concept links. If this does not give us a clear decision as to which concept to choose, then in a sense there must be little difference between them in terms of relevance, linkage to the student's existing knowledge, and intrinsic importance in the domain. In the last resort, we therefore adopt the arbitrary measure of simply choosing one of the concepts in this final list (such as the first one).

We can summarise the algorithm for choosing a new concept to be discussed as follows:

function: choose_next_concept

arguments:

system_beliefs

student_beliefs

function body:

working_memory <- seven concepts in *system_beliefs* with highest activation level

chosen_concept <- concept with highest node strength sum in working_memory

hypothesis of Anderson's (1983) theory, and we make no psychological claims for the validity of that numerical measure.

¹⁸ In effect, we are presently making the assumption that "known" can be equated with "mentioned", when this is clearly not the case. More sophisticated dialogue mechanisms for *diagnosing* knowledge of one dialogue participant by another would have to be introduced to begin to approximate the criteria which are used by speakers for deciding if other 'know' something or not.

¹⁹ This is a very approximate measure, based on standard techniques used in ITS, such as in the 'ECAL' system (Elsom-Cook & O'Malley 1989).

```

    if chosen_concept single element, return this concept as value of next_concept
    if chosen_concept list of concepts with equal highest node strength sum,
chosen_concept<- concept with highest importance rating
    if chosen_concept single item, return this as value of next concept
    if chosen_concept list of items with equal importance, return value of first item in the
    list as value of next concept

```

Choosing a new instance to discuss

Preconditions which choose an instance to discuss (if one has not already been chosen and bound to the 'inst' parameter) occur as part of the CONCRETE_CLAIM goal. Choosing a new instance to discuss is based on the same principles as choosing a concept, with the additional complication that one must consider the subconcept trees beneath the *justification types* for the justification instances for that belief instance.

The first step is therefore to *choose a concept* in an identical manner as above (the function `choose_next_concept`), with the obvious difference that one must restrict the contents of the initial working memory to those concepts with highest activation level which *possess at least one instance*. Once this initial working memory contents has been established, the function `choose_next_concept` is applied, to give us a single concept from whose set of instances we have the problem of choosing a specific instance to talk about. This procedure is in fact applied at a higher level in the dialogue goal hierarchy than the CONCRETE_CLAIM goal, in the preconditions for the CLAIM goal. At the point where we have to choose an instance, therefore, the concept has already been chosen. The factors considered are:

- If there is only a single instance, then of course, this instance is the one chosen to discuss.
- If none of the instances have justification instances, then they must all have the same set of possible general justification types²⁰, and we have therefore no principled means of choosing between them, so the first instance in the list is chosen.
- A factor to be considered, is the *number* of justification types which a specific instance possesses: an instance which has a greater number of justification types gives a greater degree of 'pedagogical leverage' in terms of the number of dialogue moves which can be pursued, and hence would be preferred in this sense.
- We need to consider the *prototypicality* of the justification types possessed by an instance²¹ - for example, if all the justifications are *subtypes* of the justification type 'chord_progression' then this gives less pedagogical leverage than an instance which also has justification instances which are subtypes of the 'contrast' justification type, since we will have a smaller *distribution* as opposed to *number* of topics to discuss.

²⁰ In the fixed representation of semantic memory possessed by the system we do not incorporate mechanisms for allowing the system to acquire new concepts from the student, which is left as an issue for further research.

²¹ We are simply aiming to incorporate some very general notion of Rosch's (1975) theory that prototypes are fundamental to concept representation, as one part of the algorithm for choosing a (prototypical) instance to discuss which may enable a student to better acquire a representation of the concept.

- When a justification instance is stated, we often need to explain the subconcepts attached to that justification type: for example, if we state that the chord progression 'chord V to chord I' is a justification for the existence of a phrase boundary, then we need to explain what a chord progression is and what chord V and chord I are. We therefore need to take into account the new concepts which would be introduced, in choosing a belief instance in terms of the justification instances which it possesses.

The overall algorithm for choosing a concept instance to discuss on the part of the system, therefore takes all these factors into account, by defining an *ordering* on these factors, and using them to successively select the preferred instance. It seems reasonable to first select at the most general level of the kinds of new justification types which would be introduced. The ordering of these factors which is preferred in selecting an instance is therefore as follows:

- (1) justification range ;
- (2) justification prototypicality ;
- (3) justification subconcept learnability.

We can summarise the algorithm as follows:

function: choose_next_inst

| chooses an instance to discuss from a list of instances for a given concept |

arguments:

instance_list # | for some concept c | #

system_beliefs

student_beliefs

function body:

instances <- instances list for concept

if

only one member in instances, return this as chosen instance

if

none of instances have justification instances, return the first member of instances as the chosen instance

else,

if single instance with greatest range, return this as the chosen instance

if a single instance has greatest prototypicality, return this as the chosen instance

if more than one instance with greatest prototypicality,

if single instance has highest sum of node strength sums for associated subconcept trees, return this as the chosen instance,

otherwise, choose the first instance in the current list of possible instances

Deciding whether concepts and instances are 'known' to a speaker

A frequent dialogue state precondition checks whether a concept is 'known' to a respondent, before an initiator makes an utterance which explains that concept. For example, a precondition to the ABSTRACT_CLAIM goal is

```
(NOT (known? c (ltm_model (dialogue_participant s))))
[i.e. 'for a current speaker, s, check if c is 'known' to the
other speaker, according to their memory model]
```

As we discussed earlier, whether a concept is 'known' or not is a matter of degree, in terms of the *node strength* associated with that concept. Again, to that extent, any fixed decision as to whether a memory representation which is assumed to have been encoded and remembered is 'known', must be an arbitrary decision: 'what numerical node strength must a node have in order to be "known" or not?'. We make the simple assumption that

if, according to the system's representation of the student' beliefs the student possesses a node corresponding to encoding of that concept, then the student possibly 'knows' that concept, otherwise the student does not 'know' the concept

This assumption is incorporated into the predicate `known?` listed in the previous example. Since our memory models depend on a simple encoding hypothesis and the assumption that memory trace strength increases with mental rehearsal, we do not consider the issue as to whether a memory trace corresponds to 'belief' or 'knowledge': these issues are embodied in the way in which the memory traces are treated in *dialogue*. Accordingly we need to define a somewhat arbitrary set of categories of 'known', 'not known' and 'possibly known' in order to address decisions made in dialogue (principally concerning teaching precursors, as discussed in §5.5.3). The set of categories used is as follows:

for any concept node, a concept is

not-known, if strength=0, or no node encoded in memory

possibly-known, if $1 \leq \text{strength} \leq 5$

known, if strength > 5

We assume that such distinctions are assumed by dialogue participants, who make assumptions concerning their desire not to tell the other participant what they already know, to tell them what they do not know, and that they are sometimes uncertain about whether the other knows something or not. The numerical measures used in KANT to characterise these facts are arbitrary and thus the importance of these distinctions lies in the effect on the *dialogue utterances generated*, as will be discussed in §5.6. If a concept is 'possibly known' (it has a relatively low node strength since it has not been mentioned often, and/or it has not been mentioned recently, so that the node strength has decayed), then the system generates *preannouncements* as part of dialogue actions, which give the student the opportunity to say something like "yes, I do know about that, and it is not necessary to continue with this topic". If

it is 'known' then the system does not attempt to teach precursory knowledge, and if it is 'not-known' the system decides to discuss the topic without preannouncement (but with the normal negotiation procedure). These features will be explained in the section on dialogue actions. Such a use of explicit dialogue utterances (such as preannouncements) to some extent avoids the problems in student modelling which would be implied by a simple encoding hypothesis in dialogue.

Summary of role of dialogue state preconditions

In summary, dialogue state preconditions fulfill the following functions in dialogues generated by KANT:

- checking whether a proposed new concept to discuss is in focus ;
- choosing new concepts and instances to discuss ;
- checking the previous dialogue history to determine whether the proposed dialogue move has a previous move to which it could relate ;
- checking whether conversants have the necessary beliefs to be able to discuss a topic, and whether respondents could reasonably understand a discussion about a topic.

The preconditions therefore play a crucial active role in dialogue generation, and fulfill many of the functions which are couched in other rule-based or procedural terms in previous Intelligent Tutoring Systems. Many of the methods used for characterising 'importance' of concepts and the extent to which they are 'known' are largely expedient in the sense of using arbitrary numerical measures, but are non-arbitrary in the sense of attempting to capture fundamental distinctions according to a psychological theory of memory. We are therefore not claiming that the preconditions which we have identified are necessarily the only or correct ones which could be identified: we have been concerned here to define the overall functional role of such preconditions in a dialogue generation system.

5.5.2 Dialogue Negotiation

A phase of *explicit negotiation* is the principal determining factor on the succession of possible dialogue moves and their topics which form the interactive tutorial dialogues generated by KANT. The actual mechanisms used are similar in principle to those used by Power (1979) , but do not arise in the context of proposing actions to achieve cooperative physical goals. In addition, we propose a 'microstructure' to negotiation phases, rather than a simple fixed announcement of goals. We shall now discuss the negotiation phase in terms of what is presently implemented, and in terms of possible features which could be implemented as further research. Metadialogue sequences of this kind clearly have their own structure, the explication of which forms an area for future research in Intelligent Teaching Dialogues.

In the present implementation, the NEGOTIATOR can be considered as a 'black box', the input and output to which is a set of *parameter values*. If the output is nil, then no parameters have been cooperatively agreed, and the negotiated dialogue move *fails*. If a set of values are output, then the goal *succeeds*, and any dialogue actions and effects are performed.

The current negotiator proposes the parameter values which are *currently bound*. For example, in the parameter set

```
((n system) (s student) (c nil) (inst nil))
```

neither concept nor instance parameters are bound. The current negotiator therefore generates a text template as follows:

```
"Do you accept that the student is the next speaker?"
```

If they had been bound, as follows,

```
((n system) (s student) (c scale) (inst nil))
```

then a text template would be generated as follows:

```
"Do you accept that the student is the next speaker about the concept 'scale'?"
```

In the present first prototype implementation, the respondent has simply the options of disagreeing or agreeing ('yes/no?'), with obvious implications for success or failure of negotiation.

5.5.2.1 The system as negotiator

The following is an example of a negotiation phase which has been recorded from an interaction with the system²², where the system is the negotiator:

```
Do you accept that we pursue the dialogue goal
CLAIM (== make a claim concerning a concept ) (y or n?)
y
Do you accept that we discuss the concept PHRASE_BOUNDARY
(y or n?)
y
Do you accept that we pursue the dialogue goal
CONCRETE_CLAIM (== make a specific claim about a concept instance)
(y or n?)
y
Do you accept that the STUDENT is the speaker
y
Do you accept that we discuss the concept instance P1 (y or n?)
```

²² The example derives from our own interaction with the system, and has not been 'edited'. For some of the examples shown in §5.6, we have removed some utterances which result from "bugs" in the program which would have been time-consuming to fix. For example, on some occasions, the dialogue action text is displayed twice in succession, rather than once.

y

Do you accept that we pursue the dialogue goal

MAKE_INSTANCE_CLAIM (== state a claim about an instance) (y or n?)

y

In the example, student responses are in bold typeface. As the system traverses the goal tree, it negotiates the parameters which are newly bound. The first comment that we might make is that if we were to make this into a model which is suitable for real use - rather than as a model execution trace - *the phase is extremely lengthy*, and somewhat cumbersome. It may be more practically useful to simply collect all bound parameters until a dialogue move is reached at the bottom of the tree which has an associated actual *dialogue action*. The above example would then reduce to

"Do you accept that THE STUDENT makes a CONCRETE_CLAIM about the instance P1 of the concept PHRASE_BOUNDARY?

y"

A problem arises, however, if we consider what happens in the event that the student responds 'n' to this example: we have no way of knowing from this response whether the student does not accept that CONCRETE_CLAIM is the goal, that STUDENT is the speaker or that PHRASE_BOUNDARY is the concept and P1 the instance, or any combination of these possibilities. It seems reasonable that what we require is a *combination* of this truncated approach initially, with an *extended finegrained negotiation phase* in the event of negotiation failure, in order to determine the precise parameter(s) concerning which negotiation has failed. The following are proposed extensions to the system - as further work - which are designed to address some of these problems:

- Upon failure of the system's negotiation in truncated form, the system reinvokes the CONVERSER, and uses the present 'finegrained' negotiation mechanism.
- Upon failure at some point, concerning a specific newly bound parameter, the system proposes alternative values, and prompts the student for a response. For example, if the system's proposal that the speaker should be STUDENT fails, the system prompts with SYSTEM as student, and so on.
- The system now reinvokes the EVALUATOR with this parameter value input by the student.
- If the dialogue state preconditions for the current goal can be satisfied with this proposed value, then the goal succeeds with this value, and the CONVERSER continues with the finegrained negotiation mechanism.
- In the event of actual goal names (eg CHALLENGE) receiving a negative response, the system proposes the alternative exclusive-or goal (eg CLAIM) and tries to satisfy the preconditions of this goal. If they can be satisfied, then the goal succeeds.
- If all alternative possible parameter configurations have been tried and failed, then the

goal fails as usual.

Similar amendments would be made with the student as negotiator, but there are also a number of additional considerations.

5.5.2.2 The student as negotiator

The following is an example of a negotiation phase with the student as negotiator. Underlying generation mechanisms remain the same - in this case we are simply *reading* in the student's input choices of dialogue moves and concept and instance goals, then *printing* the system's response (in the COMMUNICATOR) rather than *printing* the system's choices, and *reading* the student's response.

Please type who you want the SPEAKER to be ... (student or system)

student

Please type in the concept you want to talk about

(type 'show' if you want a full list of possibilities)

phrase_boundary

Please type in the instance position you want to talk about

p1

I will tell you if I agree to cooperate ...

Ok, I will cooperate

Please choose one but not both of the following goals to pursue:

CHALLENGE = make a challenge to a previous claim

CLAIM = make a claim concerning a concept

claim

I will tell you if I agree to cooperate ...

Ok, I will cooperate

Please choose one but not both of the following goals to pursue:

CONCRETE_CLAIM = make a specific claim about a concept instance

ABSTRACT_CLAIM = make a general explanatory claim about a concept

concrete_claim

I will tell you if I agree to cooperate ...

Ok, I will cooperate

Please choose one of the following goals to pursue:

MAKE_INSTANCE_CLAIM = state a claim about an instance

SUPPORT_INSTANCE_CLAIM = state the justifications for an instance

make_instance_claim

I will tell you if I agree to cooperate ...

Ok, I will cooperate

You are pursuing the dialogue goal (MAKE_INSTANCE_CLAIM state a claim about an instance), with PHRASE_BOUNDARY as the concept, and P1 as its instance Please confirm or disconfirm your intentions (y/n)

y

Ok, I'll remember that.

Please choose one of the following goals to pursue:

MAKE_INSTANCE_CLAIM = state a claim about an instance

SUPPORT_INSTANCE_CLAIM = state the justifications for an instance

(type 'done' if you don't want to choose one).

done

Again, student responses are in bold typeface. Student choices are followed by the phrases

"I will tell you if I agree to cooperate ...

Ok, I will cooperate "

which indicate when the system is checking the preconditions for the current goal with the student proposed parameters. As with the system as negotiator, the phase is extremely 'finegrained', and could be similarly reduced. In the present implementation, if the student proposes values which the system cannot use, then it simply refuses to cooperate and takes its own negotiated turn, as in the following example:

Please choose one but not both of these goals to pursue:

CONCRETE_CLAIM = make a specific claim about a concept instance

ABSTRACT_CLAIM = make a general explanatory claim about a concept

concrete_claim

I will tell you if I agree to cooperate ...

I'm sorry, but I do not agree to cooperate...

My move ...

Do you accept that we pursue the dialogue goal

CHALLENGE (== make a challenge to a previous claim) (y or n?)

<<...>>

To improve the negotiation phase in a similar manner to the system's negotiation phase, the system needs to generate textual *explanations of why it cannot cooperate*. For example, if it cannot cooperate because the proposed new concept is not in focus, it needs to generate text similar to

"I do not want to cooperate with your proposal that we discuss the concept 'KEY' at this point, because it is not relevant to the previous topic 'CONTRAST' which we discussed. I think we should discuss one of the concepts"

The following is an indication of some further work which would be required in order to include some of these features in KANT:

- modify the EVALUATOR to step through each precondition in turn as it evaluates the

precondition expression in the current environment ;

- for each precondition type, associate a separate declarative representation of a text template which explains its function (as in the above example) ;
- where a precondition fails on a student negotiated turn, generate the associated text template ;
- generate the set of possibilities which the system could satisfy (where appropriate, such as in the case of possible new foci, or alternative subgoals), and present them to the student
- if the student chooses a possibility generated by the system, or proposes a value which the EVALUATOR can satisfy, then the system accepts, otherwise not.

5.5.2.3 Negotiation, acceptance and rejection

The fact that, generally speaking, dialogues between people educated in certain cultural traditions exhibit certain regularities, does not reflect some universal grammaticality in discourse, or use of a finite set of discourse genres, but rather that given certain types of utterances, there are mutual *expectations* about what kinds of utterances may follow. These expectations may turn out to be unfulfilled and produce perfectly intelligible and natural sounding dialogues. One way in which these expectations can be characterised is in the dialogue phenomenon known as "*adjacency pairs*" (Schegloff & Sacks 1973). For example, the 'question-answer' pair

Michael: "What are you having?" ((question))

Mark: "Mine's a pint of Hook" ((answer))

One use of adjacency pairs which we have incorporated into our model of negotiation concerns the range of possible seconds to a first of an adjacency pair. Adjacency pairs do not in fact have to be adjacent, as for example in the following nested pairs, where the first response to the question-answer pair is in fact the first of another pair:

Michael: "Are you staying for another one?" ((question 1))

Mark: "Have you got enough money on you?" ((question 2))

Michael: "I think so" ((answer 2))

Mark: "OK, I'll have a Hook" ((answer 1))

It is clear that two possible seconds are to give the expected *acceptance* response, or a dispreferred *rejection*. These are incorporated into our model by providing the alternative responses 'y' or 'n', to the respondent. However, another way of 'rejecting' to play the 'dialogue game' (Levin & Moore 1977) which the adjacency pair attempts to initiate (rather than simple rejection) is to reply with the first of a quite *different* adjacency pair, as in the example above. So doing can represent an attempt by the respondent to pursue a different kind of game or goal. This is incorporated implicitly into our model by *alternation of negotiated turns*: after a respondent responds with 'n' to an initiator's attempted goal pursuit (the first of an adjacency pair), the goal is abandoned as failed (negotiation fails to secure an agreed goal), and the respondent pursues a goal using the first of a different adjacency pair in an attempt to secure negotiation preconditions of their goals. The 'rejection' may, however, be temporary, as in the example of nested adjacency pairs, since the respondent may be temporarily suspending their acceptance of the first proposed goal, initiating some other goal, then returning to subsequently accept the first speaker's re-negotiation of the first goal. In this way, the first of one adjacency

pair can receive the first of another as its (dispreferred) second. A feature which we have not incorporated into the model is this *preference ordering* between (a larger number of) possible seconds, since we presently know of no theoretically interesting way of characterising context dependent ordering.

As can be seen in the example above, where negotiation receives a negative response, negotiation phases may thus become *extended*: after negotiator n1 on negotiated turn t1 receives a negative response, this is immediately followed by turn t2, negotiator n2, with a new negotiation phase. It is an open question whether we choose to regard one negotiator's negotiation immediately followed by another as two separate phases, or as the same phase, since from the point of view of the utterances actually generated, there is no intervening dialogue action to separate them.

5.5.3 Dialogue Actions

Dialogue actions are distinguished from utterances involved in the negotiation phase by their concern with the *belief states of conversational participants*, rather than with the past, future or proposed form of the dialogue itself. These latter utterance types are termed *metadialogue sequences*. KANT deals with only *two* fundamental types of belief, concerning *concepts*, and *instances*. For the purposes of belief communication in the INFORMER, it is immaterial whether these concepts are justification types or concepts which are justified, or whether instances are instances of justifications or of concepts. A concept is the name of any node, irrespective of the link types which it possesses. However, if we state some proposition in dialogue, the previous dialogue may be important in its interpretation. For example, the statement

"... there is some blue sky ..."

may be interpreted as:

- a *statement* of a simple fact that sky which is blue exists ;
- a *justification* of a previous utterance that tomorrow the weather will be sunny;
- a *denial* of a previous statement that the sky in England is always grey;

and so on, depending on the conversation which has preceded this utterance. When KANT uses the INFORMER to communicate some belief, it therefore relies on the conversants' understanding of the *functional role* of that belief communication utterance, as defined in the previous metadialogue sequences - i.e. the negotiation phase. For example, a dialogue action which communicates

"p88 is an instance of concept 'harmony'"

could be a simple statement if preceded by

"CONCRETE_CLAIM (= state a claim about a specific instance of a concept)"

or a statement that p88 is a possible instance of a concept, given a previously stated instance with similar justification types if preceded by

"COMPLEMENTARY_CLAIM (= an instance with the same justifications as previously stated instance)"

The negotiation phase therefore fulfills this additional role of securing the future interpretation in context of dialogue actions.

The INFORMER consists principally of a function called INFORM, which has the following simple syntax:

```
(INFORM <speaker> <concept(s)> <instance(s)>)
```

Executing a dialogue action (if one exists for a goal) by the INFORMER therefore involves substituting the negotiated parameter values into this function expression, and evaluating the expression. The following are examples of substituted INFORM expressions (the functions `justifications` and `justification_types` simply evaluate to return the justification instances and types for an instance and a concept, respectively):

```
(INFORM 'system 'phrase_boundary 'p3)
[the system informs respondent that p3 is an instance of concept
'phrase_boundary']
```

```
(INFORM 'system nil ((justifications p3)))
[the system informs respondent that j11,j12,j13 are instances of
justifications for instance p3]
```

```
(INFORM 'system 'phrase_boundary nil)
[the system gives a general explanation of the concept phrase_boundary
to the respondent]
```

```
(INFORM 'system ((justification_types 'phrase_boundary))
nil)
[the system informs the respondent that jp,jq,jr are justification
types for the concept phrase_boundary]
```

```
(INFORM 'system nil 'p9)
[the system states that p9 is a instance, stated in the context that
the dialogue move being pursued is to state some complementary
instance to the instance previously stated]
```

Different procedural interpretations are provided by the `inform` function, depending on whether the given substituted arguments are

`speaker = system or student`

concept = a single concept or a list of concepts (or nil)

instance = a single instance or a list of instances (or nil) .

5.5.3.1 System INFORMing

The concept parameter will be a single element in the case where a CLAIM move is made, for the MAKE_CONCRETE_CLAIM or MAKE_ABSTRACT_CLAIM moves. In the former case, the instance parameter will also be a single element (the instance claimed) and in the latter it will be nil. The concept parameter will be a non-nil list in the case where a list of justification types is informed, as in the SUPPORT_ABSTRACT_CLAIM move (when the instance parameter will again be nil), and the instance parameter will be a non-nil list in the case where a list of justification instances is informed (in which case, the concept parameter will be nil). When any parameter is a single item, the inform function translates this with a text template, and where it is a list, it translates each item in the list with the template. The examples given above illustrate this. These simple rules represent the first level of 'informing'. We have adopted the view that since 'informing' the student about some concept involves mentioning other concepts as part of the textual explanation, the student may need to understand some of these subconcepts in order to understand the 'first level'. For each concept informed, therefore, the system generates its associated subconcept tree in the semantic network representation, and attempts to CONVERSE certain of those subconcepts. There are two points to be explained here:

- (1) the system needs some means of deciding on the order of presentation of those subconcepts for each concept informed;
- (2) the system needs some means of deciding for each possible subconcept, whether it should be CONVERSED or not.

(1) *searching subconcepts*

In the first case, we adopted the policy of generally searching the subconcept network *depth-first, left-to-right*, in a manner similar to a 'successive refinement' approach²³, but recognise that other search methods are possible here, in future versions of KANT. For example, we could include the bottom-up 'cognitive apprenticeship' method, but in the event of providing such multiple domain traversal strategies, we would need some principled way of choosing between them²⁴. In KANT the major constraint on topic shifts concerns dialogue *focus*, as characterised by the spreading activation model which we described earlier. It is therefore consistent that we should include this factor in the algorithm for traversing subconcept networks in informing. Accordingly, as the algorithm generates subnodes to be searched, these nodes are *ordered in terms of activation level*, with the most active to be explored first by the left to right method.

(2) *Deciding whether to CONVERSE a subconcept or not*

As the search algorithm reaches each individual subconcept, it needs to make a decision as to whether to CONVERSE that concept or not. For any concept in the semantic network, there is

²³ As defined in the DOMINIE system (Elsom-Cook & Spensley 1988).

²⁴ The current research most relevant to this question has been carried out by Elsom-Cook and Spensley (1988), on the 'DOMINIE' system - a 'domain independent intelligent tutoring system', with multiple teaching strategies.

usually quite a large number of subconcepts, so if all concepts were conversed, we would have an unnecessarily lengthy extension to each inform statement. If a subconcept is to be CONVERSED as part of the INFORMER, then that concept becomes an *input parameter to the CONVERSER, which may itself involve further invocations of the INFORMER*. Thus, the dialogue actions may be *recursively embedded within each other*. Suppose on exiting such a recursion, we consider conversing a concept which has already been conversed, then the decision would be negative. For example, in the subconcept network shown in figure 5.4 above, suppose we began by INFORMING the concept 'key'

(inform 'system 'key nil)

This would result initially in the text associated with 'key' being presented. Now the INFORMER considers the subconcept network beneath 'key'. Supposing that the subnode 'chord_I' is the most active of its subnodes, then the CONVERSER is reinvoked (under certain conditions) with the input parameters

((n system) (s nil) (c chord_I) (inst nil))

In other words, we now have the option of either the student or the system discussing the concept 'chord_I', with no instance currently specified. The usual negotiation and action phases would follow, which may lead to subconcepts of 'chord_I' being informed as part of this second-level 'informing'. When this is completed, the sibling nodes of 'chord_I' would be candidates for informing.

The decision as to whether to attempt to re-CONVERSE a subconcept node as part of a dialogue action is based on whether the system believes that the student *'knows'* that concept or not. It would be possible to avoid making any decision here, and apply the CONVERSER to *all subconcepts*, since the attempt to make a claim about a 'known' concept, would lead to precondition failure (the *known?* predicate) at this level. We adopted the view that this may be wasted computational effort, since known concepts could be eliminated from consideration at this stage, and that in any case there may be some special dialogue features involved in continuation of discussing one concept which are distinct from discussing a new concept. These special features have been represented as *metadialogue presequences*, which relate to the system's beliefs concerning the *extent* to which a concept is known or not. We discussed what it means for KANT for a concept to be 'known' or not in the earlier section on dialogue state preconditions. We have characterised this concept in a relatively neutral manner as simply 'remembering', which does not consider the philosophical problems associated with the definition of knowledge and belief, and its logical formalisation: 'knowing' in our terms relates to the *strength* of a memory trace, which is a function of how frequently and recently it has been explicitly mentioned. This is quite reasonable if we consider the fact that it may be relevant to remention a concept which was mentioned a relatively long time ago in the dialogue, for one participant to 'remind' another. In a computational model, we are however,

forced to make some decision, based on numerical strength values, which is (as we mentioned earlier) arbitrary in its precise value, but non-arbitrary in theoretical principle. The simple decision which we made was as follows: for any concept node, a concept is "not-known" (strength=0, or no node encoded in memory), "possibly-known" ($1 \leq \text{strength} \leq 5$) or "known" (strength > 5)²⁵. This translates into a simple decision process as to whether to attempt re-CONVERSING:

```

if concept c not-known, re-converse c
if c possibly-known, generate preannouncement
    if positive response, re-converse c, otherwise do not reconverse
if c known, do not re-converse c

```

Clearly, the system determines the truth function of these predicates by looking at its representation of the student's semantic memory. Cases where a concept is classified as *possibly-known* represent the system's uncertainty about its representation of the student's beliefs: the concept has been mentioned sufficiently frequently and recently, but not enough to be quite sure that it is remembered. In such cases, Intelligent Tutoring System's often resort to sophisticated techniques for *diagnosing* the student's knowledge. In keeping with the concept of the centrality of the interaction model in KANT, we have used the dialogue phenomenon of *preannouncements* for this purpose. Such uses of explicit dialogue, along with a simple encoding hypothesis for concepts mentioned in dialogue, to some extent pre-empt some of the problems of student diagnosis, although this difficult issue has not been a focus of this research concerning tutorial interaction.

In terms of Levinson's (1983) analysis, there is a certain class of preannouncements which can be analysed as two sets of superimposed adjacency pairs, whose structure we simplify as follows:

- 1 - checks 'newsworthiness' of potential announcement in position 3
- 2 - presequence second, *and* first part of second pair = either pre-empting of 1, or acceptance, as 'request to tell'
- 3 - second part to second pair, announcement delivered on acceptance in 2
- 4 - acknowledgement of 3

An example (fabricated) might be

- 1 - "Did you hear the news about Jim?"
- 2 - "No, what?"
- 3 - "He was injured in a car accident last night"
- 4 - "No, really...?"

or alternatively,

- 1 - "Did you hear the news about Jim?"
- 2 - "Yes, I heard he got injured"

These structures are incorporated into KANT as follows:

²⁵ See discussion in §5.5.1.2.

- 1 - system checks newsworthiness of proposed concept to be discussed ;
(it looks at its own semantic memory representation to determine whether the node strength for that concept is sufficiently low);
- 2 student responds 'no' ('don't tell me about it, because I already know'), or 'yes', ('let us continue to discuss it') ;
- 3 - in the event of acceptance in 2, system generates first of adjacency pairs involved in negotiation of concept ;
- 4 - student responds 'y/n'.

In summary, if a proposed subconcept is classified as possibly-known, the system generates preannouncements of the form:

"I would like to continue and discuss <concept list>, being subconcepts of <concept> that I think you ought to understand.
Is that ok? (y/n)"

In the event of the response 'y', the system continues and negotiates each concept, invoking the CONVERSER; otherwise it responds

"Ok, maybe we'll discuss <concept list> in a future turn. Your move ..."

5.5.3.2 Student informing

The mechanism involved in managing the student's dialogue actions (with the *student as speaker*) is a relatively trivial one: the system substitutes the negotiated and agreed parameters involved in the student's negotiated turn, utters them in the form of a text template, and asks the student to confirm or disconfirm their intentions. If the student confirms, the system utters a simple acknowledgement, otherwise the student is given the option of re-defining their negotiated turn. For example, suppose the student has successfully negotiated the parameter values

((n student) (s student) (c 'chord_I) (instance 'i666))

then the system generates

"I understand that you are claiming that i666 is an instance of the concept chord_I. Please confirm or disconfirm this (y/n).

y

Ok, I'll remember that.

My move ..."

On a student negotiated turn, if the student has asked that the *system* be the next speaker, then the mechanism for evaluating an *inform* statement such as

(inform 'system 'key nil)

for example, is carried out in an identical manner as if it had been the system which had chosen itself as speaker. There is one single difference in this case, concerning the possible informing of *subconcepts* of the initial concept concerning which the student negotiated that the system 'inform'. We can not necessarily assume that the student wants this further informing to take place, although an identical evaluation of the inform statement with the system as speaker would do this. In this case, therefore, a special pre-sequence is generated, in the form of an *offer*, which may receive *acceptance* or *rejection*. The offer concerns the system's desire to continue to discuss subconcepts. When the INFORMER reaches this point, it therefore generates a template of the form

"I want to continue to discuss some further concepts of <concept> which I think you should understand. Do you accept that? (y/n)"

Upon acceptance, the system continues the recursive application of the CONVERSER to subconcepts, with the attendant possibility of generating preannouncements for possibly-known subconcepts. These features will be exemplified in the example dialogue discussed in section 5.6.

5.5.3.3 The fundamental algorithm for the INFORMER

We can now summarise the algorithm for the inform function, evaluated as part of the INFORMER:

function: inform

arguments:

s # | student or system | #
 c # | a concept, a list of concepts or nil | #
 inst # | an instance, a list of instances or nil | #
 system_beliefs

function body:

If s is student, do student_inform with same arguments
 If s is system, do system_inform with same arguments

function: system_inform

arguments:

n # | student or system | #
 s # | student or system | #
 c # | a concept, a list of concepts or nil | #
 inst # | an instance, a list of instances or nil | #
 system_beliefs

function body:

if both c and inst are nil, return nil # | no concept or instance to INFORM | #

```

if n is system # | system negotiator & speaker | #
then
  INFORM_SUBCONCEPT_NETWORK for c,
  if c is atom and inst nil # | single concept to discuss, no instance(s) | #
    generate explanatory text for c
  if c is a list and inst is nil # | list of concepts to INFORM | #
    generate explanatory text for each member of c
  INFORM_SUBCONCEPT_NETWORK for c,
  if c is nil and inst is an atom # | only instance to state | #
    say, "I am claiming that <inst> is an instance of the concept we are
      discussing"
  if c is nil and inst is a list
    say, "I am claiming that <inst> are instances of the concept we are
      discussing"
  otherwise, return nil

if n is student # | system negotiates that student speaks | #
  say "you are claiming that <inst> is an instance of <c>."
  Please confirm or disconfirm"
  If 'yes' say "I will remember that"
  generate offer to continue with subconcepts
  if accepted, INFORM_SUBCONCEPT_NETWORK for c
  otherwise, allow repeated turn
otherwise, negotiated turn over

```

function: student_inform

arguments:

```

s      # | student or system | #
c      # | an atom, a list, or nil | #
inst   # | an atom, a list, or nil | #
*system_beliefs*

```

function body:

```

If both c and inst are non-nil, say
  "You are claiming that <inst> is an instance of <c>. Please confirm or
  disconfirm(y/n)"
If c is nil and inst non-nil, say
  "You are claiming that <inst> is an instance. Please confirm or disconfirm(y/n)"
If response is 'y' to either of above, say "Ok, I will remember that", and
  student negotiated turn is over
otherwise say "Do you want to redo your turn? (y/n)"
  if response is 'y', call student_converser
  otherwise, student negotiated turn is over

```

The auxiliary function `INFORM_SUBCONCEPT_NETWORK` attempts to `CONVERSE` (some of) the subconcepts which underlie the initial concept (and instance) `INFORMed`. It therefore incorporates a *recursive* application of the fundamental dialogue generation mechanism (the `CONVERSER`) on successive 'generations' of subconcepts of the original concept informed, involving attendant negotiation phases for each of those subconcepts, and the generation of a number of preliminary metadialogue utterances. For a given concept, the function `INFORM_SUBCONCEPT_NETWORK` expands the concept to its list of immediate subconcepts, orders the subconcepts in descending order of highest activation level, then recursively calls the `CONVERSER` with each concept in turn as an input parameter value to be discussed. This corresponds to following *local dialogue coherence* in teaching precursors (including preannouncement where the system is not sure whether the concept is 'known' or not, and the usual negotiation mechanisms). Preannouncements (as described earlier) can avoid reapplication of the `CONVERSER` when the student responds effectively that discussing the subconcept will not tell them something they do not already know. If the system believes that the subconcept is known, it does not re-converse, otherwise it simply re-converses without preannouncement, which will immediately invoke a negotiation phase in the `CONVERSER`. Recursive descent continues until negotiation fails with some concept value, in which case the student is asked if they want to continue with other possible subconcepts, or else to end the current negotiated turn. On non-acceptance, the program *exits* the informing process, and the negotiated turn is over, otherwise it may continue to `CONVERSE` other 'sibling' subconcepts. The function for discussing precursory subconcepts can therefore be summarised as follows:

function: `inform_subconcept_network`

arguments:

`concept`

`*system_beliefs*`

function body:

`generate subconcept tree for c from *system_beliefs*`

`expand c to a list of its subnodes, &`

`order list of subnodes in terms of highest activation level`

`for each node in subnode list,`

`if unknown, call CONVERSER with node as concept parameter value`

`if known, do nothing`

`if possibly-known`

`generate preannouncement for node`

`if accepted, call CONVERSER with node as concept parameter value`

`otherwise, generate offer to continue with subconcepts`

`if accepted, continue searching concept network`

`otherwise, negotiated turn over`

There are a number of problems with the procedure for searching and conversing subconcept networks which have not been resolved in the present implementation. The two principal problems are as follows:

(1) The problem of maintaining topical coherence when the depth-first search algorithm has explored a sibling subtree, and returns to explore other siblings. For example, if we inform the list of concepts (key scale), after we have finished discussing the tree beneath 'key', several utterances will have elapsed, and there is sometimes difficulty in 'knowing where we are', when the tree beneath 'scale' is explored - i.e. we have forgotten the original purpose of discussing this concept. There are some quite simple things which can be done here, one of which is to generate a sequence which reiterates the original dialogue move and concept which led to the discussing of these concepts, for example

"We were originally pursuing the goal of SUPPORTING AND ABSTRACT CLAIM, with the justification types KEY and SCALE. We now want to give further explanation for the concept of SCALE."

(2) The possibility of introducing alternative domain traversal algorithms and choosing between them, as discussed above.

5.5.3.4 Example action

The following is a short example of the system performing a dialogue action, which it attempts to extend recursively:

Example dialogue action

((negotiating conjunctive subgoal))

Do you accept that we pursue the goal SUPPORT_INSTANCE_CLAIM (y/n?)

y

((belief instance justifications informed))

I claim that CONTRAST and CHORD_PROGRESSION are justifications for the instance P1 of PHRASE_BOUNDARY.

((checks student beliefs; student requires precursory subconcepts))

I would like to continue and explain these justification types. Is that ok (y/n?)

y

((presequence since system not sure whether known))

I'm not sure if you know about CONTRAST. Do you want me to continue and tell you about it (y/n?)

y

((recursively invokes converser with c=CONTRAST))

Do you accept that I make a CLAIM about CONTRAST (y/ n?)

y

Do you accept that I make an ABSTRACT_CLAIM (y or n?)

y

MAKE_ABSTRACT_CLAIM is the dialogue goal currently being pursued

The following is a brief explanation of the concept CONTRAST

TEXT FRAME PRESENTED

some explanatory text for node CONTRAST

<<<...>>>

n

My turn over ... your move

<<<...>>>

5.5.3.5 Summary of dialogue actions

In this section we have discussed the details of how dialogue actions are performed by a program called the INFORMER. The program substitutes negotiated parameter values into an expression which is evaluated to communicate beliefs concerning concepts and instances (or lists of both of these). After initial generation of text associated with a concept, the INFORMER may recursively call the CONVERSER (i.e. the fundamental generation mechanism for a negotiated turn) on subconcepts of the original concept explained, given that preannouncements have been generated which allow the student to pre-empt this process, if (s)he believes that the future explanations will not tell them anything which they do not already know. The INFORMER calls the set of text template generation functions which comprise the COMMUNICATOR for generating text input and reading in the student's responses. When the student performs an action, the system generates a summary statement of their parameter choices, to which they are invited to agree or disagree.

5.5.4 Action Effects and Negotiation Effects

The effects of dialogue actions - which communicate beliefs - are to change the conversational participants' belief representations. We have already discussed action and negotiation effects in sections 5.4.1.4 and 5.4.1.5. Here we summarise the EFFECTOR as follows:

function: update_ltm_models

arguments:

system_beliefs

student_beliefs

c

inst

function body:

for each of *student_beliefs* and *system_beliefs*,

 apply node strength DECAY to each node, according the Anderson
 formula since a dialogue action has taken place

 check if node for c present,

if present, increase its strength by 1, since it has been explicitly mentioned
 if not present, reset memory model to contain node with 0 activation and strength = 1, if *student_beliefs*
 if inst not present in node, reset node and memory model to contain it
 give input activation of 1 unit to node c, if *student_beliefs*
 apply Anderson model for spreading activation with c as source, by recursively expanding to subnodes, & resetting according to formula, recording nodes passed through to avoid cycling

Node strength is reduced before activation is spread, and is performed as part of action effects since this is the primary mechanism for generating successive dialogue states. The system does not update its *own* beliefs with the instances and concepts stated by the student, otherwise, there would be no possibility of later CHALLENGEs, since belief sets would be the same. It updates its model of the student.

Negotiation effects take the agreed negotiated parameters for each goal, and insert them, together with the current goal name, onto the dialogue history. We include this program as part of the EFFECTOR.

5.6 Example Dialogues and their Generation

In this section we give examples of segments of interactive dialogues which have actually been generated by the present implementation of KANT. We have already given numerous examples of separate dialogue phases in the earlier discussion of negotiation and dialogue action, which should be sufficient to explain their individual structures. We shall analyse the generated examples in terms of the programs of KANT described in this chapter. Before discussing these examples, there are a number of points to be emphasised. Our research goals were to design a high-level model for generating tutorial dialogues which incorporated the design features of Negotiated Tutoring described in the previous chapter, and to implement this high-level model in an ITS architecture in order to test its completeness and consistency. We were *not* aiming to extend the model to generation of natural language which is itself adapted to the learner, nor to design an interface to the model which is robust and easily usable. Such goals are beyond the scope of the present research. In further research it may be possible to incorporate some limited natural language understanding and generation, or to eliminate some aspects of student typed input using a graphical interface construction language. The text which follows should therefore be viewed as an *interactive trace* of the execution of the model, with text which is designed primarily to show execution. The examples are chosen to illustrate different aspects of system and student negotiated turns, with alternative speaker values. In all examples, the following typographical conventions are used:

plain text - eg "My move ..." = text generated by KANT

italic text - eg 'system' = text typed to KANT by the user

bold text in double parentheses - eg '((system chooses an instance))'

= textual comments concerning operation of KANT, typed after program execution.

Examples of dialogue fragments

The actual test semantic network used in KANT is shown in Appendix A. In this example, we shall use a simple subset of that network for illustrative purposes within this chapter²⁶. In this example, we assume that the student starts the dialogue with some beliefs about concepts and instances.

Example 1: negotiation & dialogue goal pursuit

```
((start of dialogue; SYSTEM = current negotiator))
((satisfies s=system in goal=DISCUSS))
My move ...
((start of negotiation phase))
Do you accept that SYSTEM is the next speaker?
y
Do you accept that I make a CLAIM concerning some concept (y/n?)
((claim goal preconditions satisfied, since at start of
dialogue no claims to be challenged; negotiate with current
speaker value))
y
Do you accept that we discuss the concept PHRASE_BOUNDARY (y/ n?)
((concept parameter intially unbound; system chooses a
binding; phrase_boundary chosen because only concept with
instances to discuss, derived from parser, and no activation
inputs to date))
y
Do you accept that I make a CONCRETE_CLAIM about a specific instance of
PHRASE_BOUNDARY (y/n?)
((satisfies goal precondition evaluation since
PHRASE_BOUNDARY has instances))
y
Do you accept that we discuss the instance P1 (y/n?)
((system chooses instance because it has most prototypical
justifications))
y
Do you accept that I now MAKE_INSTANCE_CLAIM (y/n?)
y
```

²⁶ This is done for implementational reasons, since the Macintosh Plus computer upon which KANT was implemented did not have enough memory to enable the 'real' network to be searched repeatedly in Lisp, within a reasonable time without interrupting the dialogue with extensive garbage collection. Hence the examples shown are intended to illustrate the kinds of dialogue structures which can be generated, and do not directly correspond to the network shown in appendix A.

((negotiation effects propagated))

The following is a brief explanation of the concept PHRASE_BOUNDARY

((performs dialogue goal action))

TEXT FRAME PRESENTED

explanatory text for node phrase_boundary

((no text represented in system - just concerned that
'something' should be generated at this point))

Please remember for our future discussion that I am claiming that P1 is
an instance of concept PHRASE_BOUNDARY

((action effects propagated; updates conversants' belief
models to put p1 as instance in student's representation of
this node; gives concept node input activation))

((dialogue continues))

Example 2 follows on from example 1, where the initial phase of 'informing' an instance is
followed by the conjunctive subgoal of supporting it, and recursive invocation of the
CONVERSER.

----- Example 2: system INFORMS justification types

((negotiating conjunctive subgoal))

Do you accept that we pursue the goal SUPPORT_INSTANCE_CLAIM (y/n?)

y

((belief instance justifications informed))

I claim that CONTRAST and CHORD_PROGRESSION are justifications for the
instance P1 of PHRASE_BOUNDARY.

((checks student beliefs; student requires precursory
subconcepts))

I'm not sure if you know about CONTRAST. Do you want me to continue and
tell you about it (y/n?)

((presequence since system not sure whether known))

y

((recursively invokes converser with c=CONTRAST))

((new negotiation phase begins))

Do you accept that we I make a CLAIM about CONTRAST (y/ n?)

y

Do you accept that I make an ABSTRACT_CLAIM (y or n?)

((abstract_claim is satisfied because system has no instances
about node CONTRAST))

y

MAKE_ABSTRACT_CLAIM is the dialogue goal currently being pursued

The following is a brief explanation of the concept CONTRAST

TEXT FRAME PRESENTED

some explanatory text for node CONTRAST

((action and negotiation effects propagated))

<<<...section omitted for continued recursive explanation...>>>

n

((eventually student responds 'n' to proposed subconcept))

My turn over ... your move

((end of system negotiated turn))

In example 3 we see the student as negotiator, in effect requesting the system to tell the student about some concept.

Example 3: student negotiator/system speaker

((student = negotiator))

Your move. ... Do you want to continue the discussion? (y/n)

((in top level loop of ARGUMENT CONTROLLER))

y

Please type who you want the SPEAKER to be (student or system)

system

((student negotiates speaker=system))

Please type in the concept you want to talk about

contrast

Please type in the instance to be discussed

nil

((student does not want to know about a specific instance, so will receive a general explanation))

((system checks if it could satisfy goal preconds with these parameter bindings))

I will tell you if I agree to cooperate ... Ok, I will cooperate

((EVALUATOR called & succeeds))

Please choose one of the goals to pursue:

CHALLENGE = make a challenge to a previous claim

CLAIM = make a claim concerning a concept

claim

I will tell you if I agree to cooperate ... Ok, I will cooperate

Please choose one of the goals to pursue:

CONCRETE_CLAIM = make a specific claim about an instance

ABSTRACT_CLAIM = make a general claim about a concept

abstract_claim

I will tell you if I agree to cooperate ... Ok, I will cooperate

Please choose one of the goals to pursue:

MAKE_ABSTRACT_CLAIM = make an abstract claim

SUPPORT_ABSTRACT_CLAIM = talk about kinds of supporting evidence for a concepts

make_abstract_claim

I will tell you if I agree to cooperate ... Ok, I will cooperate

The following is a brief explanation of the concept CONTRAST

((system performs dialogue action of making abstract claim))

TEXT FRAME PRESENTED

some explanatory text for node contrast

((system tries to continue action with precursory subconcepts))

I would like to continue and discuss the (HARMONY_CONTRAST INTERVAL_LEAP) , being subconcepts of CONTRAST that I think you ought to understand. Is that ok? (y/n)

<<...>>

My move ...

((recursively entering CONVERSER))

In example 4 there are two negotiated turns: in the second turn the system challenges the student's claim made in the first turn.

Example 4: the system challenges a student claim

My turn over ... your move

((beginning student negotiated turn))

Are you sure you want to continue the discussion? (y or n)

y

Please type who you want the SPEAKER to be ...(student or system)

((defining speaker parameter in DISCUSS goal))

student

Please type in the concept you want to talk about (if you want a full list of possibilities, type 'show')

((system manages student negotiated turn, searching goal tree; defining concept parameter in goal DISCUSS))

phrase_boundary

Please type in the instance position you want to talk about

p10

I will tell you if I agree to cooperate ... Ok, I will cooperate

Please choose one but not both of the goals to pursue:

CHALLENGE = make a challenge to a previous claim

CLAIM = make a claim concerning a concept

((subgoals to DISCUSS))

claim

I will tell you if I agree to cooperate ... Ok, I will cooperate

((system calls EVALUATOR with input parameters;
phrase_boundary in working memory))

Please choose one but not both of the goals to pursue:

CONCRETE_CLAIM = make a specific claim about a concept instance

ABSTRACT_CLAIM = make a general explanatory claim about a concept

concrete_claim

I will tell you if I agree to cooperate ... Ok, I will cooperate

((concrete_claim satisfiable because inst chosen))

Please choose a goal to pursue:

MAKE_INSTANCE_CLAIM = state a claim about an instance

SUPPORT_INSTANCE_CLAIM = state justifications for an instance claim

make_instance_claim

I will tell you if I agree to cooperate ... Ok, I will cooperate

You are pursuing the dialogue goal MAKE_INSTANCE_CLAIM, with
PHRASE_BOUNDARY as a concept, and P10 as its instance. Please confirm or
disconfirm your intentions (y/n)

((INFORMER called, with student as speaker; summary statement
of chosen parameters generated))

y

Ok, I'll remember that.

Please choose a goal to pursue:

SUPPORT_INSTANCE_CLAIM = state justifications for an instance claim

((subgoals were conjunctive, so student can choose second of
conjunction))

support_instance_claim

Please type in the justification(s) for the instance P10

(CONTRAST PARALLELISM)

I will tell you if I agree to cooperate ... Ok, I will cooperate

You are pursuing the dialogue goal SUPPORT_INSTANCE_CLAIM, with
PHRASE_BOUNDARY as a concept, P10 as its instance, and (CONTRAST
PARALLELISM) as the justifications. Please confirm or disconfirm your
intentions (y/n)

y

Are you sure you want to continue the discussion? (y or n)

((negotiated turn over))

y

My move ...

```
((system negotiated turn - top level loop of ARGUMENT
  CONTROLLER))
```

Do you accept that we pursue the dialogue goal CHALLENGE (== make a challenge to a previous claim) (y/n?)

```
((both CLAIM and CHALLENGE could be satisfied, but
educational principles prefer CHALLENGE - satisfied because
instance stated which can be challenged))
```

y

Do you accept that we discuss the concept PHRASE_BOUNDARY (y/n?)

```
((c bound to c of instance challenged from dialogue history))
```

y

Do you accept that we discuss the concept instance P10 (y/n?)

```
((inst bound to instance challenged from dialogue history))
```

y

Do you accept that we pursue the dialogue goal CLAIM_SHARED (== agree with a previous claim) (y or n?)

y

I agree that P10 is an instance of the concept PHRASE_BOUNDARY

```
((here the negotiation of CLAIM_SHARED in effect tells the
student that the system agrees with the instance))
```

Do you accept that we pursue the dialogue goal AGREE_SUPPORTS (== agree with some previously stated justification types) (y or n?)

y

AGREE_SUPPORTS is the dialogue goal currently being pursued with the justification type(s) PARALLELISM.

```
((inst now bound to set of supports for challenged instance
from dialogue history, which was (CONTRAST PARALLELISM).
PARALLELISM is a member of system's justification set for
this instance))
```

I want to try to explain what I mean by the justification(s) PARALLELISM. Is that ok? (y/n)

```
((INFORMER called to generate previous utterance for
AGREE_SUPPORTS; now checks student belief representation and
finds PARALLELISM possibly-known, so recursively calls
CONVERSER))
```

n

Ok, maybe we'll talk about PARALLELISM next time around...

```
((control returns to CONVERSER at level of traversing goal
operators))
```

Do you accept that we pursue the dialogue goal NEW_SUPPORTS (== state some justifications which we not mentioned for previous instance claim)

(y or n?)

y

NEW_SUPPORTS is the dialogue goal currently being pursued with CHORD_PROGRESSION as the new support(s). I want to try to explain what I mean by the justification(s) CHORD_PROGRESSION. Is that ok? (y/n)

n

Ok, maybe we'll talk about CHORD_PROGRESSION next time around...

((as for previous goal, INFORMER calls CONVERSE; student does not accept))

Do you accept that we pursue the dialogue goal

DISAGREE_SUPPORTS (== disagree with some previously stated justification types) (y/n?)

y

DISAGREE_SUPPORTS is the dialogue goal currently being pursued with CONTRAST as the disagreed support(s).

I want to try to explain what I mean by the justification(s) CONTRAST. Is that ok? (y/n)

n

Ok, maybe we'll talk about CONTRAST next time around...

My turn over ... your move

((system negotiated turn over: CONVERSE returned nil since no more subgoals to current goal))

In example 5, the system initially cooperates, but then finally refuses to cooperate - because the student has proposed a topic which the system does not regard as being sufficiently in focus.

Example 5: the system does not cooperate with a student negotiation

My turn over ... your move

Are you sure you want to continue the discussion? (y or n)

y

Please type who you want the SPEAKER to be ...(student or system)

system

Please type in the concept you want to talk about (if you want a full list of possibilities, type 'show')

key

Please type in the instance position you want to talk about

nil

please type 'ok' (without the quotes) when you're done

ok

I will tell you if I agree to cooperate ...

Ok, I will cooperate

Please choose one but not both of the goals to pursue:

CHALLENGE = make a challenge to a previous claim

```

CLAIM = make a claim concerning a concept
claim
Please type in the instance position you want to talk about
nil
I will tell you if I agree to cooperate ...
Sorry, I do not cooperate ...
((at this point system calls EVALUATOR with parameters
  ((n student) (s system) (c key) (inst nil))
dialogue state precondition evaluation fails in CLAIM goal
since concept 'key' is not sufficiently in focus - i.e. its
activation level is too low.
Student negotiated turn over))
Are you sure you want to continue the discussion? (y or n)
y
My move ...

```

These examples of portions of dialogues with KANT have been intended to show a variety of features, including student and system negotiated turns, the effects of student and system negotiation acceptance and rejection, and CLAIM and CHALLENGE branches of the dialogue moves represented. Some features have not been shown, which are included in the specified algorithms and their extensions for KANT, such as 'the system negotiating that the student should speak'. We shall summarise these features which represent the difference between the *algorithms - i.e. the specification of KANT* - and the present implementation in the next section.

5.7 Extensions to the present implementation towards the present specification

In this section we shall briefly summarise some of the extensions to the present implementation of KANT, which would be required in order to extend the implementation towards the current *specification* of Negotiated Tutoring, which was motivated in the previous chapter. We summarise these features mentioned in the text, together with some new features which will be discussed in the next chapter.

Allowing speakers to challenge their own previous claims, and representing additional subgoals for challenging in this case

In the present implementation speakers can not challenge their own previous claims. In section 5.5.1.1 we discussed extensions to permit this, including an additional subgoal for challenging in which a conversant is not required to either agree or disagree with the actual claim. The extension represents an alteration to dialogue state preconditions to remove this restriction.

Implementing the system CONVERSER procedure of initially attempting to satisfy student as speaker

From the examples shown in the previous section, it can be seen that the procedure of

attempting to satisfy dialogue state preconditions with the *student* as speaker when the system is negotiator has not yet been implemented. This involves a modification of the EVALUATOR, algorithms for doing which have been described in section 5.4.1.3.

Implementing truncated negotiation phase generation, and re-negotiation with new parameters

In section 5.5.2 we discussed the negotiation phase in detail, and indicated how the NEGOTIATOR could be modified to included a 'truncated' summary of parameters, which reverted to the present 'fine-grained' version upon negotiation failure, and recalled the EVALUATOR to check whether alternative proposed parameters could be satisfied. This would only provide an initial indication as to how negotiation could be managed in tutorial dialogues. The precise structure of such explicit negotiation metadialogue phases, and the incorporation of some degree of *implicit* negotiation forms of major goal of further research.

Generating new presequences to improve coherence in subconcept search

As part of subconcept teaching, the dialogue loses topical coherence when a depth-first recursion returns to a higher level node, or when the INFORMER discusses a list of concepts: the student loses track of the original purpose of explaining that concept. We suggested that as an initial amendment, the system could generate additional presequences at these points, which restated the original goal, and the purpose of the next preferred teaching interaction.

Extension of algorithms for concept and instance choice to gain pedagogical leverage from student's conception of dialogue focus

The present algorithms for choosing a new concept and instance focus are largely based on looking at the amount, relevance (focus-based) and prototypicality of new subconcept material to be introduced. Future versions could decide to be more 'student centred' in the sense of considering which concepts would seem most relevant from the *student's* perspective in terms of activation levels in the representation of the student's beliefs. This would involve altering the dialogue state precondition which checks focus, so that it is applied to the memory representation of the proposed speaker rather than negotiator.

Inclusion of mechanism in EVALUATOR for checking individual precondition failures, and attempting to 'repair' this by generating explanations to students

In the present implementation, the system's reasons for not cooperating with the student's negotiation are presently opaque to the student. In section 5.5.2 we suggested how the EVALUATOR could be modified to check and evaluate each precondition in turn, to generate textual explanations of what the failed precondition(s) mean(s), and to propose possible parameter values under which the precondition could possibly be satisfied. This is one example of 'negotiation repair' which will be discussed in further work in the next chapter.

More fundamental issues for future research are described in the 'further work' section of chapter 6. They are viewed as 'fundamental' in the sense that we believe that progress in

addressing these issues may involve fundamental changes in the architecture of the system which we have described. Before having pursued these directions for further research, we can not, of course, substantiate this belief.

5.8 Summary of Chapter 5

This chapter was concerned with giving an explanation of the algorithms involved in the current specification of the Intelligent Tutoring System KANT, and in defining the extent to which the current implementation meets these specifications. Descriptions were given of how the implementation could be so modified. Where required, relationships with research which was not described in chapter 3 were described, such as the use of Anderson's (1983) model of semantic memory, and the use of adjacency pairs and presequences in KANT. As we stated at the end of chapter 2, KANT and GRAF (the model for musical structures) were not fully integrated in the research reported in this thesis. We therefore adopted the approach of basing KANT on a knowledge representation which we assume to be output by GRAF. The principal distinctive features of KANT are:

- 1 its use of spreading activation and associated memory models for characterising focus in tutorial dialogues, and for modelling the changing course of participants' beliefs ;
- 2 the use of explicit negotiation as a means for controlling the succession of dialogue units in interactive tutorial dialogues ;
- 3 the use of 'dialogue state preconditions' for defining a set of possible utterances by student and system, and of a set of educational principles for choosing between them where there are multiple possibilities; and
- 4 the representation of teaching styles in terms of a set of fundamental dialogue moves.

In the next chapter we shall critically review these and other features of KANT, in comparison with the literature reviewed in chapter 3, and describe major areas for future research.

Chapter 6: Conclusions, critique and further work

"In accordance with reason's legislative prescriptions, our diverse modes of knowledge must not be permitted to be a mere rhapsody, but must form a system. Only so can they further the essential ends of reason. By a system I understand the unity of the manifold modes of knowledge under one idea."

(Immanuel Kant, Critique of Pure Reason 1781, § 'The Architectonic of Pure Reason', trans. Kemp-Smith 1929)

6.0 Introduction

In this final chapter we shall draw together conclusions concerning research contributions to ITS and contributing disciplines, which were described in the individual chapters, followed by a *critique* of these contributions with respect to some very specific areas of existing work which have been reviewed in separate sections of the thesis. We shall also make some new conclusions. Finally we shall describe how the research could be continued as further work.

6.1 Critique

In this section we discuss research contributions in relation to the most relevant areas of research discussed in this thesis, in order to assess precisely the extent to which our knowledge has been extended, and to describe limitations of the research, some of which will be addressed by further work. Once we have briefly stated the most closely related areas of research, the next section summarises research contributions.

6.1.1 A cognitive model for musical structures: critique

GRAF reformulates and extends the work of Lerdahl & Jackendoff (1983), and thus most closely relates to their work. We have described the problems with the Lerdahl and Jackendoff theory in detail in chapter 2, which principally concern its general *informal* nature, the psychological implausibility of any processing mechanisms posited, the lack of a model of interrelating processes, the formalisation of musical parallelism, and a failure to distinguish multiple levels of perceptual and cognitive processing. These are the principal issues - many of which were raised by Lerdahl and Jackendoff - to which GRAF provides some initial answers. GRAF has present limitations in terms of severe domain restrictions (which limit its educational application), and certainly does not succeed in incorporating *all* of the subtle musical intuitions in Lerdahl and Jackendoff's theory.

The most closely related work which followed or was based on the Lerdahl and Jackendoff theory is that of Ebcioglu (1986), Heefer & Leman (1986), Ashley (1988) and Jones, Scarborough and Miller (1988). All of these workers adopt a variant of an expert-systems approach (including blackboard architectures and multiple production systems), and thus have problems in terms of their putative status as cognitive models. An advantage of the approach of Ashley

(1988) is some notion of the role of learning in musical perception, which we discussed as a possible extension of the grammatical formalism of components of GRAF in chapter 2. Ebcioglu's (1986) research provides a powerful formalisation of the reductional theories of Schenker (1932), as an alternative to the Lerdahl and Jackendoff time-span reduction theory. Schenker's work is now beginning to be questioned in musicology (Narmour 1977), and Ebcioglu's model represents this knowledge in the form of a large set of production rules, which do not inform us about the processes involved in perception of reductions (the approach involves chronological backtracking) nor the strategies performed by human analysts.

6.1.2 KANT and the Negotiated Tutoring approach: critique

Negotiated Tutoring draws on a large number of areas of ITS and dialogue research. We shall restrict our discussion to a relatively small subset of the most closely related research.

Negotiated Tutoring as a general approach to ITS

No existing ITS research is based on an explicit statement of a tutorial approach equivalent to Negotiated Tutoring, which emphasises symmetry, negotiation and multiple interaction styles, although each individual component has closely related areas of research, as we discuss in the sections which follow.

The most closely related paradigm, which has previously stressed the primacy of the interaction model is the *Guided-Discovery approach* (Elsom-Cook 1984). One of the primary distinctive features of Guided-Discovery Tutoring (in addition to the primacy of interaction models) is the conception of ITS approaches as points along a dimension of the degree of *freedom* or *constraint* on the student in the tutorial interaction (Elsom-Cook 1984). We can thus see Negotiated Tutoring as taking a position along this dimension which treats the student as an equal participant in the interaction, in terms of the range of interaction goals available to them, and their freedom to negotiate them and influence the overall course of their learning. Negotiated Tutoring can therefore be viewed as a form of a Guided-Discovery approach which aims to specify the mechanisms which can provide this freedom in the interaction, with the system's decisions as to whether to cooperate or not (in the dialogue negotiation phase) as a representation which enables us to alter flexibly this degree of constraint. A further feature of Guided-Discovery Tutoring is the idea that the tutor should aim to *successively withdraw* its guidance towards a situation of free discovery, once the student has demonstrated sufficient ability to explore the domain with a sufficient degree of directedness. In Negotiated Tutoring these two factors are related: if we give sufficient freedom to the student in the interaction, then they should be able to withdraw from receiving the system's guidance themselves. Since they are given sufficient freedom to pursue their own negotiated turns, then they are similarly given the freedom to terminate the interaction¹. As we discuss in the sections which follow, many of the elements of Negotiated Tutoring are presently implemented in limited and crude forms, their refinement being future research.

¹ Since the symmetrical provision of dialogue moves and mechanisms for negotiating their use does not necessarily imply their use, nor cooperation with what is being negotiated, there is thus no necessary conflict between "symmetry" in the interaction and "successive withdrawal". In other words, the fact that a conversant accords symmetrical dialogue possibilities to another does not necessarily imply that they will always comply with the use of those possibilities.

Interaction symmetry

In chapter 4 we stated that the Negotiated Tutoring approach to symmetry in ITS involves the goal of providing (as far as presently feasible) a set of interaction styles for each dialogue participant. In KANT this is achieved by providing the student with *almost* the same set of interaction goals as the system. Few recent approaches have adopted this general goal in ITS, earlier research being mostly confined to "mixed-initiative" dialogues which allowed the student to ask questions (Collins & Stevens 1983; Clancey 1987). A recent approach which has also stressed the need for a move beyond tutorial interaction as a "one-way street" is the work of Farrell (1988) on the 'DECIDER' system and its "self-education" approach. The major difference between DECIDER and KANT/Negotiated Tutoring is that the former does not propose symmetry within an explicit model of the interaction, i.e. *it does not propose a model of dialogue*. In fundamental terms the interactions with DECIDER correspond to a set of questions and answers, reminiscent of Socratic Tutoring approaches, with sophisticated modelling techniques for matching those answers with the system's prototypes, and responding to point out exceptions: there are none of the features commonly associated with models of rational agency and high-level dialogue which are present in simple forms in KANT.

The principal respect in which KANT does *not* succeed in its goal of interaction symmetry is in allowing the student to freely express their conceptualisation of the domain. This is because KANT did not consider the considerable problem of providing an unrestricted natural language 'front end', and did not have a knowledge-base which could be rationally extensible in response to the student's statement of an area of knowledge, or a different conceptualisation of the same knowledge, which was beyond the scope of its own. These problems are reasonably beyond the scope of the present thesis, involving fundamental issues of machine learning, natural language understanding and modelling of a student's idiosyncratic conceptualisation of a domain. As such they remain some of the fundamental problems for all ITS research. In effect, this problem implies that the initial assumption which we made of a complete separation between "high-level dialogue" decisions, and 'low-level' linguistic utterances, can not be completely retained. The Negotiated Tutoring approach nevertheless provides us with a functional architecture within which these difficult problems can be addressed. We anticipate that inclusion of conceptual recognition into the system may have important effects on the range of dialogue moves which need to be provided for diagnosing and *negotiating* the meaning of utterances. The general goal of interaction symmetry has also recently been proposed in the theoretical model of 'Intelligent Tutoring Dialogue' proposed by Petrie-Brown (1989). She expresses this goal as follows:

"We should build generative models of dialogue addressing the intentions of both participants in the interaction rather than only applying analytic models concerned with the purpose on one interactant."

(Petrie-Brown 1989, p. 195).

Negotiation and cooperation in tutorial dialogues

The only work of relevance to the use of negotiation in Negotiated Tutoring is to be found in dialogue research *per se* - no existing work in ITS has adopted negotiation as an explicitly conceived phase in tutorial dialogues for determining future cooperative teaching and learning goals. The most closely related dialogue research (which we have reviewed in chapter 6) is that of Levin & Moore (1977), Power (1979), and Frolich (1988). Levin and Moore provided indications as to how a theory of implicit negotiation might be developed. The model of implicit negotiation which they propose fundamentally relies on partial matching of elements of utterances with dialogue game parameters - for example, if a speaker *s1* refers to concept *c1* and speaker *s2*, and we know that *s1* does not know *c1*, then it is a plausible inference that *s1* is asking *s2* for information (INFO-SEEK) about *c1* (alternatively, they could be asking for HELP about *c1*, or a number of other possibilities). Formalising implicit negotiation is a hard unsolved research problem, which we leave as a major focus of further work. KANT has no natural language understanding facilities, and so can not engage in linguistic analysis of input sentences in an attempt to partially match them with its set of dialogue moves. Since our research focus was on the role of negotiation in a wider model of intelligent tutoring, we have initially adopted the approach of explicit announcement of negotiated dialogue moves, after the work of Power (1979). This makes the negotiation phases as implemented in KANT somewhat cumbersome, although we have made a preliminary specification of some improvements to the generated structure (see chapter 5).

A further aspect of dialogue which needs to be addressed by implicit negotiation are certain real-time aspects of dialogue, and a set of assumptions for 'turn-taking'. The work of Frolich (1988) on explanation dialogues includes some of these aspects - such as presumption that the present speaker 'holds the floor' unless interrupted - based on the work of Schegloff & Sacks (1973), which need to be incorporated into KANT. A further influence on the cooperative style of Negotiated Tutoring also comes from the field of explanation, and the work of O'Malley (1987) on cooperative explanations. An important feature of such dialogues is the negotiation of a common point of view on the domain. KANT assumes that both participants will adopt the same point of view on a domain, and interaction mechanisms for expressing it. The negotiation of such multiple viewpoints on knowledge (Moyse 1989) is an important area which Negotiated Tutoring needs to address.

Critical argument and justified belief

KANT implements a set of dialogue moves for critical argument as an appropriate approach to tutoring with justified belief. Negotiated Tutoring is not equivalent to critical argument, since the approach could logically be retained with other sets of dialogue moves. We do not, therefore, view critical argument as a kind of separable 'discourse genre' - in any case, the "critical arguments" of KANT integrate dialogue moves for argument, critique, instruction and explanation, since they are *arguments with an educational purpose*. Other relevant work in argument generation, expert systems and ITS is that of Collins & Stevens (1983), Miller (1984), Stutt (1987) and Reichman (1985). The *dialogue moves* used in KANT are closely related to those used by *all* of this work: the essential difference consists in the *dialogue model* within

which they are represented (if there is such a model). None of these approaches use negotiation as the fundamental mechanism for controlling interaction element transitions: we would argue that such a mechanism enables us to better model the structure of dialogue (Schegloff & Sacks 1973; Levin & Moore 1977; Frolich 1988), whilst recognising that tutoring dialogue may involve some different elements to that of human conversations (Petrie-Brown 1989).

Teaching styles and educational principles

Little existing work has made the explicit connection between work on teaching styles and work in dialogue found in KANT and Negotiated Tutoring. Petrie-Brown (1989) has recently emphasised the necessity to consider the teaching interaction as dialogue in a general sense, and some recent work by Elsom-Cook (1989b, in press) has begun to represent Socratic teaching styles in terms of logical dialogue primitives. The requirement of providing multiple teaching styles and using an education principle ('least directedness') as a mechanism for choosing between them was first approached in the DOMINIE system (Elsom-Cook & Spensley 1988). In DOMINIE, interaction styles are restricted to those of the *teacher*, and teaching styles are represented as fixed conventional pairings of single interaction styles with domain traversal algorithms. There is no explicit discussion of the necessity to provide flexible combinations of these two factors. The educational principles used in KANT are influenced by the empirical work of Bellack et al (1966) and the conception of "progressive" teaching approaches described in Bennett (1976).

Topic and intention in tutorial dialogues

In KANT a specific approach is adopted to the relationship between intentional and topic-based structures. Other relevant work which has approached this problem is Levin & Moore (1977), Appelt (1982), Elsom-Cook (1984) and Grosz & Sidner (1986). A use of spreading activation in dialogue focus, similar to that used in KANT, was proposed by Levin and Moore (1977). Ours is the first use of such a model in tutorial dialogues, and the specific uses of the approach in KANT and in Levin and Moore's work are different: in their work spreading activation is used for partial matching of dialogue games in dialogue comprehension, whereas in KANT, spreading activation controls cooperative assent to proposed topics in terms of their *relevance*, and is used as one of a number of criteria for choosing new topics to discuss. In an ITS context, the work of Elsom-Cook (1984) was the first to suggest that topical and intentional structures need to be integrated in tutorial dialogues. He adopted a different approach to control of topic shifts and choice of interaction style. Appelt's (1982) work is not concerned with dialogue *per se*, but has important implications for the relationship between knowledge and intention in language generation. He would claim that no simple separation can be made between deciding what to say and how to say it, both mechanisms being integrated at various levels within a hierarchical planner. In addition, he adopts a sophisticated formalism for reasoning about agents' knowledge and beliefs, based on the belief logic of Moore (1980) and a possible worlds semantics. In KANT the two structures are integrated within intentional structures, and the decision as to what to say and how to say it is integrated within preconditions for use of intentional structures with respect to the dialogue state, and their

subsequent negotiation. Since it does not address the issue of natural language generation, lower-level linguistic constraints can not be applied to the dialogue generation process. This was a deliberate design feature of KANT, since we wanted a model of high-level dialogue which is relatively independent of the medium of expression of higher-level decisions (Kiss 1986). For the purposes of demonstrating (or 'tracing') the model's execution we have used simple text generation techniques, for which we make no pedagogical claims as to their usefulness nor adaptation to the student. The techniques used for reasoning about belief used in KANT are very rudimentary, and are embedded in (Lisp) procedures. Future research should concentrate on a more principled declarative representation of these procedures, to enable the incorporation of the kind of reasoning techniques used by Appelt (and others, such as Cohen & Levesque 1987). The method used in KANT is, however, based on a psychological theory of semantic memory (Anderson 1983) - assumptions are made about belief encoding and retrieval - and Appelt's work is considered from a non-psychological, AI point of view². An important feature of the theory of attention and intention in language being developed by Grosz & Sidner (1986) is the notion of the primacy of intentional structures - attentional structure is represented within intentional structures, and provides constraints on intentional shifts on a goal stack. KANT adopts a similar point of view, in that dialogue moves control the shifts in topic, and local topical coherence is in turn used to constrain possible dialogue move transitions.

Planning and opportunism in dialogue

The form of opportunism in tutorial dialogues incorporated into KANT has no precedents in the ITS literature, although Petrie-Brown (1989) has made similar proposals. Existing work which incorporates intentional structures usually formulates them into *plans* (Cohen & Perrault 1980; Peachey & McCalla 1986; Cawsey 1989), which presents problems for generation of symmetrical and fully interactive dialogues. We have argued that our approach needs to be complemented by some conception of *partial planning*, which aims to satisfy multiple simultaneous goals of both dialogue participants, and retains the possibility of genuine cooperativity and interactivity. Similar proposals for a role of partial planning are also being made by Elsom-Cook (1989a). The form of opportunism which we have proposed and implemented can be viewed as closer to the ideas of Suchman (1985) on 'situated action', applied to tutorial dialogues, in the sense that we propose an alternative to extended planning which does not reduce to unprincipled opportunism.

6.2 Research contributions

In chapter 1 we argued for a conception of ITS research as essentially interdisciplinary, with a major function of *synthesising* research in related fields. The new synthesis in turn has contributions to make to those contributing disciplines. We have attempted to synthesise a number of existing trends in ITS research under a general approach called Negotiated Tutoring. Our implementation of some aspects of the approach draws on, and has implications for, work

² This is not a *criticism* of Appelt's work, since in fact he makes no psychological claims. His system uses a hierarchical planner which makes no attempt to constrain the direction of processing, nor any kind of working memory, and no psychological claims are made for the logical system used for reasoning with beliefs (nor its possible-worlds semantics). We are not, of course, denying that the formal methods which he describes could make a contribution to a future psychological model for sentence planning.

in the fields of education theory, computational linguistics, intelligent tutoring systems, and areas of artificial intelligence such as logical theories of rational agents, and approaches to explanation and argument. This research also has implications for research in music education, music and the cognitive sciences and musicology. In this field, clear boundaries are not easily drawn; we shall summarise contributions under the convenient headings of contributing disciplines, but there will naturally be some degree of overlap.

6.2.1 ITS research: contributions

Contributions to ITS research have been made in the design and/or partial implementation of KANT and the general Negotiated Tutoring approach.

Negotiated Tutoring as a general approach to ITS

We have defined an approach to Intelligent Tutoring called Negotiated Tutoring, as a synthesis of some current ideas in the field. Its primary research contribution consists in its *synthesis* of existing trends in ITS research and the indication of how aspects of these trends may begin to be brought together within a single implementation of an ITS. We have instantiated the approach within a specific domain - musical structures - which makes the approach particularly pertinent, in terms of its characteristics of justified belief, incompleteness and existence of multiple solutions. The approach is, however, generalisable to other domains, particularly but not necessarily those which share the common cognitive characteristics of our example domain. The question of the applicability and educational effectiveness of the approach, as measured with respect to the educational goals of any particular philosophy of education, is an *empirical* question (Bennett 1976), which we do not address in this thesis. *Isolated elements* of some of the major components of Negotiated Tutoring - interaction symmetry, negotiation and cooperation, multiple teaching styles and dialogue units, cognitive and metacognitive skill acquisition - can be found in contributing research in dialogue, explanation, and to some extent, in ITS (as can be seen from the critique in section 6.1.3). *No existing ITS approach aims to synthesise all of these elements.* We shall discuss specific contributions of elements of the approach in following sections.

Negotiation in tutorial dialogues

The principal controlling mechanism on the tutorial interaction in Negotiated Tutoring is a phase of (explicit) negotiation designed to ensure cooperativity once interaction opportunities are given to each dialogue participant. This is a key component of Negotiated Tutoring, the extension and formalisation of which will form a major focus of future research. Although the necessity for such a dialogue phase is apparent from research in computational models of dialogue (Levin & Moore 1977; Power 1979), especially in an implicit form, to date no existing research has applied this approach to ITS and generation of tutorial dialogues. Negotiation provides a new conception of the relationship between successive interaction units in tutorial dialogues - other existing approaches have adopted grammars (Reichman 1985), transitions defined solely by connections between domain units (Collins & Stevens 1983) and plans (Peachey & McCalla 1986; Woolf 1988).

Critical argument and justified belief

Most existing research in ITS has applied a pre-existing domain formalism, which is usually assumed to be complete and certain, leading to a highly directive tutorial style which aims primarily to 'communicate knowledge' (Wenger 1987). Recent approaches to ITS are beginning to seriously question this fundamental assumption (Farrell 1988; Seely-Brown 1989; Self 1989) and often emphasise knowledge as (socially) constructed by the learner. Where a knowledge base is viewed as possibly 'uncertain' there is a current of research in expert-systems applied to ITS which view this uncertainty as a major problem for tutoring, and attempt to eradicate it by formal AI methods - such as mathematical or statistical approaches to reasoning with uncertainty (Cohen 1985) - largely motivated by an attempt to retain the directive strategies which would be conventionally paired with tutoring 'certain' knowledge. A quite different approach to domains where it would be difficult - even if it was preferable - to represent knowledge for tutoring is the 'extended environment' approach, largely inspired as a descendent of LOGO (Papert 1980). In simple terms we can therefore see two general approaches which have previously been adopted to tutoring with uncertain knowledge³: the first is the attempt to reduce that uncertainty, and the second is to eschew a 'tutoring' approach, allowing the student to explore an environment. Our approach to this problem is quite different in that we have attempted to create *interaction styles* which are appropriate to such domains, without attempting to 'reduce away' their cognitive characteristics, nor leaving the student to explore without guidance. Existing approaches to argument generation in a tutorial context (Farrell 1988; Cawsey 1989) do not integrate argument within an explicit model of dialogue. We would therefore claim that this research contributes to ITS research in terms of showing how critical arguments may be integrated within more general models of tutorial dialogue, and how they may present a set of interaction styles which are appropriate to intelligent tutoring with justified belief.

Teaching styles, educational principles and tutorial dialogues

The educational and ITS literature on teaching styles (see Elsom-Cook & Spensley 1988; Goodyear 1989) and work on computational models of dialogue have not been explicitly connected within the ITS literature. Furthermore, we find terms like 'educational approach', 'teaching style', and 'teaching strategy' used almost interchangeably, with few clear definitions of terms. We would argue that at some level, teaching styles must be represented as combinations of dialogue units, with specific educational goals, and that the two areas should be subsumed within a general model of the *interaction*. At present, few workers in ITS have made this connection explicitly, and work on dialogue in ITS is a small but growing field (Elsom-Cook 1984, 1989; Baker 1989; Cawsey 1989; Petrie-Brown 1989). We have made a preliminary description of how multiple teaching styles may be represented as constituent dialogue units, and how the two may be integrated within a general model of the interaction, thus making a close connection between the areas of teaching strategies research and dialogue research.

³ We prefer the term 'justified belief' for the domain representation of KANT.

Topic and intention in tutorial dialogues

Any approach to (tutorial) dialogue generation must explicitly or implicitly adopt some point of view on the interrelationship between domain knowledge and conversational participants' goals, and how this is used to drive the dialogue forward. In dialogue research (Grosz & Sidner 1986) this is usually referred to as the relationship between intention ('goals') and attention ('topic' conceived as 'subject'). We have specified a different approach to this interrelationship in terms of four interacting processes which control tutorial dialogue generation, as follows:

- (i) possible intentional units are satisfied with respect to a *dialogue state* (participants' beliefs and a dialogue history);
- (ii) a spreading activation model of dialogue focus both controls local focus shifts and suggests new possible concepts and beliefs to discuss with respect to specific intentional units;
- (iii) a set of educational principles prefer certain intentional units over others;
- (iv) negotiation mechanisms control whether the intentional unit can be cooperatively pursued.

Components of the model can be found in existing dialogue research (see the 'critique' section of this chapter); but the model as implemented in KANT, specifies a *new relationship* between such controlling mechanisms for generation of tutorial dialogues. Although most of these elements exist in dialogue research, *this combination of spreading activation models of dialogue focus, explicit negotiation and educational principles has not been applied to generation of tutorial dialogues.*

6.2.2 Education theory: contributions

The results of empirical educational research and educational philosophy are difficult to apply to ITS simply because those fields have not asked the same kinds of questions as ITS, nor have they sought to answer them with similar precision and formalisability. Elsom-Cook (1989) has argued for a role of ITS as a 'test-bed' for the formalisation of theories of educational interaction. The KANT model therefore provides a framework within which to approach the issues of formalisation of teaching and learning styles, and the elucidation of educational principles which prefer certain styles over others. In addition, the Negotiated Tutoring approach forces us to precisely specify the meaning of such terms as 'teaching style', 'teaching strategy', 'educational approach' and so on.

6.2.3 The cognitive science of music: contributions

Negotiated Tutoring was implemented in the specific domain of teaching musical structures in given examples of melodies, and we therefore developed a cognitive model for perception of musical structures which has made a number of contributions to other fields apart from ITS. The GRAF model makes independent contributions to the specialised field of Music and the Cognitive Sciences⁴.

⁴ The model has been presented at recent major conferences and symposia in the field (Second International Conference on Science and Music, City University, London 1987; First International Symposium on Music and the Cognitive Sciences, IRCAM, Paris, 1988), and has been published in two papers in consecutive issues of the journal *Contemporary Music Review* (Baker 1989a, 1989b).

Formalising Lerdahl & Jackendoff (1983)

Lerdahl and Jackendoff's (1983) "generative" theory of music has formed the basis of music work in experimental psychology of music (Deliege 1987; Todd 1989), yet practitioners state that its *informality* poses problems in making clear testable predictions (Clarke 1986). We have shown how Lerdahl & Jackendoff's (1983) informal theory of music cognition may be reformulated, formalised and extended to produce a theory for perception of musical structures, which integrates the theory of schematic recognition used in the experimental psychology of music. We have produced a model of underlying cognitive processes in a field where existing models are largely informal and descriptive.

Modelling processing in music cognition

Existing models of music cognition concentrate on a single aspect - such as harmonisation or 'unconscious inference' of metre - and do not specify how these component processes interrelate. GRAF describes how the processes of pattern perception, musical reductions, surface grouping features, unconscious inference of harmony and schematic recognition in terms of symmetrical grouping structures, may *interrelate* in music cognition. The model reformulates the Lerdahl and Jackendoff well-formedness rules and 'normal forms' as high level schemata, and the grouping preference rules as surface boundary detectors. Further research would be required to incorporate existing work on perception of key and metre.

Levels of processing in music cognition

Lerdahl and Jackendoff's theory is essentially 'flat' in that it does not distinguish successive levels of cognitive and perceptual processing. Subsequent work which aimed to formalise the theory similarly failed to distinguish low level perceptual 'feature detection' and 'pattern recognition' processes from higher level schematic recognition. GRAF represents these levels of processing as low-level surface boundary detection, and matching with high-level grouping schemata via intermediating resource limited harmonic processing. Although the model describes in general terms how such levels can be interrelated, future versions should represent surface grouping features at the *acoustical* level.

6.2.4 Computational musicology: contributions

Existing approaches to musical analysis (Schenker 1932; Namour 1977), and the time-span reductions of Lerdahl & Jackendoff (1983), rely strongly on the human analyst's musical intuitions in the approach of 'reducing' tonal and atonal pieces of music to fundamental harmonic, melodic and rhythmic events. We would argue that a computational model of reductions would enable these intuitions to be fully formalised, thus assisting in the clarification of theories in musicology. There are no existing formal models for automatically performing musical reductions: existing models use the computer as a tool to assist the analyst (Smoliar 1980), and do not formalise his/her intuitions. An exception is the Schenkerian analysis model of Ebcioglu (1986), but this model makes no claims to being integrated within a cognitive theory of how musical structures are perceived. AGA gives an indication of how the theory of musical reductions in Lerdahl and Jackendoff's theory of time-span reductions may be

formalised, in an initially simplified way. Further research in the directions indicated would be required to do full justice to the subtlety of their approach.

6.3 Further work

This thesis is concerned with an approach to Intelligent Tutoring called Negotiated Tutoring. The principal area of further work therefore concerns the formalisation, extension and implementation of that approach. In addition, we shall briefly discuss further work on the further specification and re-implementation of GRAF and KANT which has been described in more detail in individual chapters. Due to time constraints, we implemented sufficient aspects of KANT to demonstrate the feasibility of the approach, and specified some remaining aspects which were held to not change the proposed architecture fundamentally. The union of implementation and remaining specified aspects is termed the *full specification* of KANT. We therefore distinguish between further work towards the full specification and further work on the Negotiated Tutoring approach itself, of which KANT is a partial implementation.

6.3.1 The cognitive model for musical structures: further work

GRAF provides a model for music cognition which integrates a number of interacting processes on different 'levels'. It thus provides a promising basis for extension to include existing work on further processes towards an 'integrated' model of music cognition.

Towards an integrated model of music cognition

Our perception and cognition of music is determined by multiple factors - our 'unconscious inference' of harmony, of grouping and motivic structures, of recurring musical patterns, and of rhythm. For example, our perception of harmony clearly interacts with the independent perception of phrase groupings, since one indicating factor of such structures are specific harmonic sequences called *cadences*. Research has made some progress in understanding each of these perceptual processes (Lerdahl & Jackendoff 1983; Longuet-Higgins & Lee 1983; Winograd 1968), and the GRAF system described in chapter 2 has described how a number of these processes (grouping, harmony and reductions) interact within a cognitive model. In order to develop *unified* models of music cognition we need to look to general theories in cognitive science now that a number of individual processes in music have begun to be described. This problem is particularly important at this stage in research in music and the cognitive sciences, since individual processes could be specified in ways which could not be integrated with other research. A possible direction for future research in music and the cognitive sciences is therefore to develop unifying theoretical models for music cognition, which integrate existing research into models for individual processes with applications and extensions of existing unified models of cognition. One such unified theory is the "SOAR" cognitive architecture (Laird, Newell & Rosenbloom 1987), which models 'chunking' in human cognitive problem solving. It consists of a series of connected 'search spaces', with algorithms for learning paths (problem solutions) through the spaces. Aspects of understanding a piece of music can similarly be viewed as such a search problem (Baker 1988a; 1989a), amongst possible harmonic, rhythmic and other interpretations for each successive set of pitches which we hear in a melody, with a set of context-based constraints and heuristics. The advantage of using the SOAR architecture would

be that it addresses many of the problems stated above, since it provides a *unified representational framework*, and includes *models of learning*⁵. We therefore propose that a possible (and difficult) future research direction is to combine work on grouping structure, harmony and reductions (Baker 1989a, 1989b) with work on perception of rhythm (Longuet-Higgins & Lee 1983) within the unified model of SOAR.

Extension of reductions mechanism

The current model of musical reductions incorporated in GRAF does not incorporate all of the features of the Lerdahl and Jackendoff (1983) theory of time-span reductions⁶. We concentrated instead on the general form of an implemented model, which could initially formalise the approach, leaving further details for further work. The present mechanism concentrates on harmonic features alone, and does not take a number of aspects into account, such as rhythmic and melodic emphasis in reductions, the complicating factor of the possibility of elision (see chapter 2), the interaction of 'prolongational reduction' (relating to fluctuations in tension and relaxation in the music) and certain special features for retaining fundamental 'normal form' structures at cadences. The incorporation of these features could be addressed by further work.

Extension to other genres

GRAF is presently severely restricted to a 'toy' domain, and faces the problem of its extension to other musical genres in terms of the large-scale knowledge engineering enterprise involved, in common with most models in AI. An interesting further problem is modelling the identification of 'musical context': after hearing the first few bars of a piece it is likely that listeners are able to quickly identify the likely melodic and harmonic 'vocabulary' of a piece from matching a 'context space' in long-term musical memory, and are able to use this knowledge as a *constraint* on future cognitive processing (provided that the genre of the piece is one with which they are already familiar). The attempt to model this phenomenon would form a fascinating area of future research, involving fundamental issues of musical knowledge representation and musical perception.

6.3.2 Extending KANT towards Negotiated Tutoring: further work

In this section we shall briefly summarise some extensions to the present implementation of KANT, required as further research towards the *full specification*. These extensions are discussed in more detail in individual sections of chapter 5.

Representation of dialogue history

For purposes of understanding a record of the dialogue, the dialogue history presently records all dialogue moves, rather than containing a more psychologically realistic model where conversants only record some previous statements in long term memory, and recent moves in working memory. The representation clearly depends on the extent to which it can be utilised

⁵ Ashley (1988) and Marsden & Pople (1989) have recently proposed musical learning as a central current research problem.

⁶ In personal communication at IRCAM in 1988 Lerdahl kindly pointed out a number of features which should be retained.

by the set of dialogue moves represented.

Allowing speakers to challenge their own previous claims

In the present implementation speakers can not challenge their own previous claims, and we have discussed extensions to permit this, including an additional subgoal for challenging in which a conversant is not required to either agree or disagree with the actual claim. The extension represents an alteration to dialogue state preconditions to remove this restriction.

Representation of greater and more principled range of dialogue moves

The present range of dialogue moves represented is very small, and was only intended to demonstrate the overall feasibility of the Negotiated Tutoring approach. In future implementations we shall try to incorporate additional moves for other forms of reasoning (such as 'analogical reasoning', and so on), and to expand the generation of 'canned' text for abstract claims into a more complete set of subgoals for *explanation*. This extension could build on the work of Weiner (1980) and Cawsey (1989).

Completing implementation of system CONVERSER

The procedure of attempting to satisfy dialogue state preconditions with the *student* as speaker when the system is negotiator has not yet been implemented. This would involve a modification of the EVALUATOR, and some algorithms for doing this have been described in chapter 5.

Implementing improved negotiation phase

In chapter 5 we indicated how the NEGOTIATOR could be modified to include a 'truncated' summary of parameters, which reverted to the present 'fine-grained' version upon negotiation failure, and recalled the EVALUATOR to check whether alternative proposed parameters could be satisfied. Modelling the precise structure of such explicit negotiation metadialogue phases, and the incorporation of some degree of *implicit* negotiation, form major goals of further research.

Independent declarative representation of educational principles

At present the educational principles used to choose between alternative satisfiable dialogue moves are few, and are embedded in procedures. They should be given an independent extensible declarative representation, to give greater flexibility to the system, provided that further principles of this kind can be derived.

Incorporating further domain traversal algorithms

As part of the INFORMER, subconcept trees are searched using a simple depth-first search algorithm, exploring subnodes guided by the heuristic of exploring nodes which are most in focus (i.e. active) first. Other domain traversal algorithms for intelligent tutoring have been identified in the literature (Elsom-Cook & Spensley 1988), and future versions of KANT may try to incorporate these, together with a principled decision procedure for choosing between them. We wish to retain a separation between domain traversal algorithm and interaction

elements in order to provide the possibility of their flexible combination in ways which do not necessarily embody existing conceptions of 'teaching styles'.

Improving dialogue coherence

As part of subconcept teaching, the dialogue loses topical coherence when a depth-first recursion returns to a higher level node, or when the INFORMER discusses a list of concepts: the student loses track of the original purpose of explaining that concept. We suggested that as an initial amendment, the system could generate additional presequences at these points, which restated the original goal, and the purpose of the next preferred teaching interaction. The work of Frolich (1988) has established a number of other meta-dialogue features which would need to be included.

Extension of algorithms for concept and instance choice

The present algorithms for choosing a new concept and instance focus are largely based on looking at the amount, relevance (focus-based) and prototypicality of new subconcept material to be introduced. Future versions could decide to be more 'student centred' in the sense of considering which concepts would seem most relevant from the *student's* perspective in terms of activation levels in the representation of the student's beliefs.

Improving the dialogue state precondition representation

In the present implementation, the system's reasons for not cooperating with the student's negotiation are presently opaque to the student. In chapter 5 we suggested how the EVALUATOR could be modified to check and evaluate each precondition in turn, to generate textual explanations of what the failed precondition(s) mean(s), and to propose possible parameter values under which the precondition could possibly be satisfied. This is one example of 'negotiation repair' (Douglas 1988).

6.3.3 Integrating GRAF and KANT: further work

Since the GRAF prototype (chapter 2) was not carried through to complete implementation, further work which would be required to link the system with KANT (we specified the input and output behaviour of both models).

Giving a learning perspective on system beliefs

The initial representation of system beliefs in KANT lacks a coherent perspective on the *origin* of these beliefs, in terms of the long term memory representations for concepts which interrelate beliefs derived from the musical parser. We have presently adopted the simplification that beliefs are initialised with 0 activation and a node strength of 1, when this would not necessarily be the case. Successive use of the same belief set on different dialogues may remedy this to some extent.

Use of multiple consistent belief sets

In chapter 2 we described how beliefs concerning possible phrase boundaries are represented as input to KANT from the musical parser GRAF. In fact, the individual beliefs returned by the

parser may be grouped into several *consistent belief sets*, each of which derives from a different parent grouping structure frame. For example, the phrase boundaries after positions 4,6,8,12,16 in a melody may be grouped as instances in separate consistent belief sets as follows: (p4,p8,p12,p16) (p6,p12,p18). If, for example, a student asserts p6, they are therefore in a sense committed to claiming the second of these consistent belief sets, and if they subsequently assert p4, for example, this 'contradicts' their first claim. In the present implementation we have not considered these issues, and have initially assumed that only one possible belief set will be present. If future versions of KANT included representations of such consistent sets, we would be able to represent a further subgoal of CHALLENGEing, which we might term 'COMPETING_CLAIM'. Its preconditions would relate to the possibility of the present claim asserting instances in belief set s2 when previous instances were in a different belief set s1. Such a dialogue move would give the student the possibility of revising either the present or previous claims.

6.3.4 The Negotiated Tutoring approach: further work

We now describe further *long-term* research directions to extend the goals of the Negotiated Tutoring approach itself.

Dialogue symmetry

In the present implementation there is an asymmetry in the range of dialogue moves provided for the student, in terms of *abstract claims*. KANT has no means of understanding unrestricted textual explanations for the meaning of concepts provided by the student, and no rational means of incorporating such explanations into the system's beliefs. A possible means for understanding textual input by the student within the restriction of the *system's* conceptualisation of the domain would be to use existing knowledge acquisition tools. This would not, however, address the machine learning problem of understanding and responding to the student's conceptualisation of the domain. This represents a long-term research problem for *any* Intelligent Tutoring system, and human-computer interaction research in general.

Teaching styles, educational principles and dialogue

A major component of Negotiated Tutoring has been the attempt to represent teaching styles in terms of dialogue units. The literature on teaching styles presently shows a lack of clarity with respect to the meaning and role of such terms as 'teaching style', 'educational principle', 'teaching approach', 'teaching strategy'. In chapter 4 we gave a preliminary analysis of such terms. We propose to conduct further fundamental research into the logical definition of these styles in terms of dialogue units, using analysis of teaching dialogue transcripts.

Agents, intentions and beliefs

KANT and Negotiated Tutoring incorporate an implicit model of rational agents, their beliefs, and ways of reasoning about them. The model is, however, largely embedded in procedures, and has not been made sufficiently explicit. For example, 'wants', 'desires' and 'intentions' are not represented explicitly in KANT (they are embedded in dialogue control mechanisms), to enable the application of recent sophisticated techniques for reasoning with intention and belief (Moore 1980; Appelt 1982). A major area of future research will be to apply some recent areas of

agent theory (Kiss 1986; Cohen & Levesque 1987; Elsom-Cook 1989b) to the explicit representation of intentions and beliefs, thus enabling the incorporation of more sophisticated methods of reasoning with them. Our goal would be to incorporate such features whilst retaining the kind of psychological model of memory embodied in KANT.

Planning and opportunism

The type of opportunism represented in KANT needs to be complemented by *partial plans* and planning to achieve multiple simultaneous goals in dialogue. An initial set of 'desiderata' for a dialogue model which incorporates such features has been described in Elsom-Cook (1989a).

Reasoning and dialogue

There is a difference between reasoning as a mental process, and the kind of reasoning which takes place in *dialogue*, which we term *critical argument*. One might say that the 'Socratic' teaching method is a kind of argumentation, but it does not as such teach the student to argue in the same way as the system. To date, there has been little ITS research which aims to explore this relationship between reasoning as a mental process and argument in dialogue. Edwards and Mercer (1987) discuss this relationship in terms of a contrast between Piagetian and Vygotskian views on the development of logical thinking, and the role language. They cite the work of Walderkine (1982), and argue that

"...it demonstrates the essentially social and discursive basis of learning mathematics (and, in a similar vein, logical reasoning), while retaining the importance of practical actions."
(Edwards & Mercer 1987, p. 22).

One of the goals of KANT is to teach the cognitive skill of reasoning-in-dialogue by generation of critical arguments and engaging the student in such dialogues. The problem from an ITS perspective which aims to concentrate on reason and argument, is that we presently have few sufficiently formal theories of psychologically plausible human reasoning, of argumentation in dialogue, and of the relationship between the two. One general future research goal will therefore be to explore the way in which the expectations and 'educational ground rules' of tutorial dialogues influence underlying reasoning processes in the way in which Edwards and Mercer describe. If we can flexibly represent such rules as an independent set of educational principles in KANT, then the system provides us with a tool to conduct experiments concerning the effect on performance on reasoning tasks (such as simple mathematical tasks) of tutorial dialogues with differing educational principles, and with differing degrees to which these assumptions are made explicit to the learner. This forms a long-term goal of future research.

Implicit negotiation

An important outstanding research problem, which has not been explicitly addressed in the ITS literature, is the development of tutorial dialogues which employ *implicit negotiation mechanisms*. We propose to explore the possibility of incorporating such mechanisms into KANT by extending and incorporating research on the inference of *intention* in dialogue (Levin & Moore 1977; Sidner 1985; Grosz & Sidner 1986; Allen & Perrault 1980). As we described at the

end of chapter 4, the computational recognition of intention in utterances has proved to be a very hard problem in AI and computational linguistics. Recognition of intention alone is not sufficient for implicit negotiation: we also need mechanisms for recognition of cooperative assent and *repair mechanisms* (Douglas 1988), which should be integrated within a specification of the 'microstructure' of negotiation phases. In addition the negotiation of a shared 'viewpoint' (Moyse 1989) on the domain is an important issue which we have not addressed.

Applications to explanation dialogues

A number of researchers have suggested that *explanation* in complex computing environments and second generation expert systems requires *negotiation* concerning the exact nature of the problem to be explained, and of the 'level' or content of the explanation given (O'Malley 1987; Rymazewski 1987). We propose that the Negotiated Tutoring approach could therefore be applied to the problem of developing computational models for sustaining negotiated cooperative tutorial explanations (Cawsey 1989) to provide intelligent interactive help facilities. In nearly all such complex environments, more or less extensive help files exist. In terms of building on existing features, and suggesting methods for their rational extension, we suggest that if the semantics of such files can be represented (Law 1986), then KANT can form the basis of a new and useful application in 'intelligent help systems' (Winkels, Breuker & Sandberg 1988).

Formalisation and testing of Negotiated Tutoring

Finally, we intend to fully formalise and test the Negotiated Tutoring approach. In terms of formalisation, we shall attempt to use formal logics for belief, intention and other mentalistic concepts (derived from "Agent Theory"). In terms of testing the theory, we shall analyse dialogue protocols in terms of the structures proposed by our theory, and attempt to use the analyses to develop a micro and macro theory of the structure of negotiation phases, during the course of a dialogue. Initially, we shall use available dialogues for teaching PROLOG (computer-mediated) and in physics teaching. At the end of chapter 4 we indicated how Negotiated Tutoring could be reduced to a small set of principles, some of which closely relate to the 'cooperative principles' of Grice (1975). If the fundamental principles of Negotiated Tutoring could be logically formalised, then this would assist in a more clear and falsifiable statement of the theory. In chapter 1 we stated that there are presently few ('proto')theories of tutorial dialogue with which to approach the analysis of tutorial dialogue protocols. Now that Negotiated Tutoring has been developed as an initial hypothesis, we can begin to test its predictive and analytical power on real protocols. We would concur with Petrie-Brown (1989) that 'intelligent teaching dialogues' may involve radically different kinds of structures from 'natural' human-human spoken conversations, but that we may want to include a number of such naturalistic features in our models. It is important to emphasise that we are *not* proposing the empirical evaluation of KANT itself, as a specific implemented program in a computer-based intelligent tutoring system, but rather the testing of a formally stated *theory of intelligent tutoring dialogues* (i.e. Negotiated Tutoring) with respect to human-computer and human-human (computer-mediated or not) dialogue protocols.

6.4 Conclusion

From the research reported here we conclude that our thesis stated at the beginning of chapter 1 has been demonstrated. Our thesis is that

Negotiated Tutoring is a model for the generation of teaching interactions in Intelligent Tutoring Systems, which indicates how some aspects of recent trends in intelligent teaching dialogue and human-computer dialogue research may be implemented in computer programs.

Work towards demonstrating this thesis in ITS has made a number of research contributions to a number of other disciplines - including music education, cognitive models of music, computational musicology, dialogue and human-computer interaction research - in accordance with the general approach and point of view on ITS research stated in chapter 1. Although we did not complete computer implementation of the musical domain model and its integration with our dialogue model, we based our dialogue model on a representation of justified beliefs in semantic memory which would theoretically be output from the domain model. To that extent, our dialogue model is specifically designed to address domains which *can* be so represented.

Negotiated Tutoring succeeds in integrating a number of recent trends towards more egalitarian tutorial dialogues, in terms of the increased symmetry in the interactions generated, the freedom provided to students to take an active part in their own learning and the provision of multiple interaction styles, together with mechanisms for securing cooperativity in tutorial dialogues. Only the very beginnings of the approach have been indicated, and many of the elements currently exist in crude forms in a partial implementation of the approach. Our research thus forms the basis of a future research programme to attempt to computationally formalise the goals of Negotiated Tutoring.

References

- Alber, J.W. & Hasher, L. (1983). Is Memory Schematic? Psychological Bulletin, 93 (2), 203-231.
- Allen, J. & Perrault, C.R. (1980). Analyzing Intention in Utterances. Artificial Intelligence 15, 143-178.
- Alpay, L. (1987). A Re-implementation, Extension and Evaluation of the medical expert system PIP. MSc thesis September 1987, Cognitive Studies Department, University of Sussex.
- Alvarado, S.J., Dyer, M.G., & Flowers, M. (1986). Editorial comprehension in OpED through argument units. Proceedings of the Fifth National Conference on Artificial Intelligence (AAAI-86), pp. 250-256, Philadelphia.
- Anderson, J. (1983). The Architecture of Cognition. Harvard University Press.
- Appelt, D. E. (1982). Planning Natural Language Utterances to Satisfy Multiple Goals. Technical Note 259, SRI International, California 94025.
- Appelt, D.E. (1985). Planning English Referring Expressions. Artificial Intelligence 26, 1-33.
- Ashley, R. (1988). An integrated model of musical listening and learning. Paper presented at the International Symposium on Music and the Cognitive Sciences, IRCAM, Centre Georges Pompidou, Paris, March 14-18, 1988.
- Austin, J.L. (1962). How to do things with words. Oxford University Press.
- Baker, M.J. (1988a). Tutoring with Incomplete and Uncertain Knowledge. International Journal of Educational Research, Vol. 12, 821-838.
- Baker, M.J. (1988b). An architecture of an intelligent tutoring system for musical structure and interpretation. Proceedings of the First International Workshop on Artificial Intelligence and Music, AAAI '88, St.Paul Minnesota (USA).
- Baker, M.J. (1989a). An artificial intelligence approach to musical grouping analysis. Contemporary Music Review, 3 (1), 43-68, Harwood Academic Publishers.
- Baker, M.J. (1989b, in press) A computational model for perception of musical grouping structures. Contemporary Music Review 3 (2) Harwood Academic Publishers.
- Baker, M.J. (1989c in press). Arguing with the tutor: a model for tutorial dialogue in uncertain knowledge domains. In Guided-Discovery Tutoring, (ed.) Elsom-Cook, M., Paul Chapman Publishing, London.
- Baker, M.J. (1989d). A model for tutorial dialogues based on critical argument. Artificial Intelligence and Education, (Eds.) Bierman, D., Breuker, J. & Sandberg, J., IOS Publishers, Amsterdam.
- Baker, M., Burstein, M.H & Collins, A. (1987). Implementing a Model of Human Plausible Reasoning, Proceedings of IJCAI 1987.
- Bartlett, F.C. (1932). Remembering: A Study in Experimental and Social Psychology. Cambridge University Press.
- Bellack, A.A., Kliebard, H.M., Hyman, R.T. & Smith, F.L. (1966). The Language of the Classroom. Teachers College Press, Columbia University, New York.

- Bennett, N. (1976). Teaching Styles and Pupil Progress. Open Books, London.
- Bharucha, J.J. (1987). MUSACT: A Connectionist Model of Musical Harmony. Unpublished research paper, Dartmouth College, USA.
- Bobrow, D.G., Kaplan, R.M., Kay, M., Norman, D.A., Thompson, H. & Winograd, T. (1977). GUS, A Frame-Driven Dialog System. Artificial Intelligence 8 (1977), 155-173.
- Boltz, M. & Reiss Jones, M. (1986). Does Rule Recursion Make Melodies Easier to Reproduce? If Not, What Does? . Cognitive Psychology 18, 389-431.
- Brayshaw, M. & Eisenstadt, M. (1989). A Practical Tracer for Prolog. HCRL Technical Report No. 42, February 1989, The Open University, Great Britain.
- Broadbent, D.A. (1985) A Question of Levels: Comment on McClelland and Rumelhart. Journal of Experimental Psychology:General 1985, vol.114, no.2, 189-192.
- Burton, R.R. (1976). Semantically-centered parsing. Doctoral Dissertation, University of California, Irvine, California.
- Burton, R.R. (1982). Diagnosing bugs in a simple procedural skill. In Intelligent Tutoring Systems, (eds.) Sleeman, D. & Brown, J.S., Academic Press.
- Burton, R.R. & Brown, J.S. (1976). A Tutoring and Student Modelling Paradigm for Gaming Environments. Proceedings for the Symposium on Computer Science and Education, Anaheim, California, Feb. 1976.
- Burton, R.R. & Brown, J.S. (1982). An investigation of computer coaching for informal learning activities. In Intelligent Tutoring Systems, (eds.) Sleeman, D. & Brown, J.S., Academic Press.
- Brown, J.S., Burton, R.R., & De Kleer, J. (1982). Pedagogical, natural language and knowledge engineering techniques in SOPHIE I, II and III. In Intelligent Tutoring Systems, (eds.) Sleeman, D. & Brown, J.S., Academic Press.
- Carbonell, J.R. (1970). AI in CAI: an artificial intelligence approach to computer-assisted instruction. IEEE Transactions on Man-Machine Systems, vol. 11, no.4, pp. 190-202.
- Cawsey, A. (1988). Generating Tutorial Explanatory Dialogues. Proceedings of 4th Alvey Workshop on Explanation, University of Manchester, September 1988.
- Cawsey, A. (1989). The structure of tutorial discourse. Artificial Intelligence and Education, (Eds.) Bierman, D., Breuker, J. & Sandberg, J., IOS Publishers, Amsterdam.
- Cerri, S.A., Elsom-Cook, M. & Leoncini, M. (1988). TRILL: The Rather Intelligent Little Lisper. CITE Report 48, The Open University (GB).
- Charniak, E. (1978). On the Use of Framed Knowledge in Language Comprehension. Artificial Intelligence 11 (1978), 225-226.
- Charniak, E. (1982) Context Recognition in Language Comprehension. In Lehnert, W.G. & Ringle, M.H. (editors) Strategies for Natural Language Processing, Lawrence Erlbaum.
- Charniak, E., Reisbeck, C. & McDermott, D.V. (1980). Artificial Intelligence Programming, Lawrence Erlbaum.
- Clancey, W. (1982). Tutoring rules for guiding a case method dialogue. Intelligent Tutoring Systems, (eds.) Sleeman, D. & Brown, J.S., Academic Press.
- Clancey, W. (1987). Knowledge-Based Tutoring: The GUIDON Program. MIT Press.

- Clancey, W. (1988). The role of qualitative models in instruction. In Self, J. (ed.) Artificial Intelligence and Human Learning, Chapman & Hall, London.
- Clarke, E. (1985). Structure and Expression in Rhythmic Performance. Musical Structure and Cognition (eds.) Howell, .P, Cross, I. & West, D. Academic Press.
- Clarke,E. (1986) Theory, Analysis and the Psychology of Music: A Critical Evaluation of Lerdahl,F. and Jackendoff,R. A Generative Theory of Tonal Music. Psychology of Music, 14, 3-16.
- Cohen, G. (1983). The Psychology of Cognition. Academic Press, London.
- Cohen, G.,Eysenck,M.W. & Le Voi,M. (1986). Memory: a cognitive approach. Open University Press, Milton Keynes, England.
- Cohen, P.R. & Perrault, C.R. (1979). Elements of a Plan-Based Theory of Speech Acts. Cognitive Science 3, 177-212.
- Cohen, P.R. & Levesque, H.J. (1987). Persistence, Intention, and Commitment. Technical Report 415, SRI International, California.
- Cohen, P.R. (1985). Heuristic Reasoning about Uncertainty: An Artificial Intelligence Approach, Pitman, London.
- Cohen, R. (1987). Analyzing the Structure of Argumentative Discourse. Computational Linguistics, Vol. 13, Nos. 1-2, January-June 1987, pp. 11-24.
- Collins, A. Warnock, E.H. & Passafume, J.J. (1974). Analysis and Synthesis of Tutorial Dialogues. BBN Report No. 2789, Boston, Mass.
- Collins, A. & Loftus, E.F. (1975). A Spreading Activation Theory of Semantic Processing. Psychological Review, 82, 407-428.
- Collins, A. (1977). Processes in Acquiring Knowledge. Schooling and The Acquisition of Knowledge, (eds.) Anderson, R.C., Spiro, R.J. & Montague, W.E., Lawrence Erlbaum, Hillsdale New Jersey.
- Collins, A. & Stevens, A. L. (1983). A Cognitive Theory of Inquiry Teaching. In Reigeluth, C. M. (ed.), Instructional-Design Theories and Models: An Overview of their Current Status. Lawrence Erlbaum Associates, Hillsdale New Jersey.
- Collins, A. & Smith, E. (1988). Readings in Cognitive Science. Morgan Kaufmann Inc., Palo Alto, California (USA).
- Deliege, I. (1987). Grouping Conditions in Listening to Music: An Approach to Lerdahl & Jackendoff's Grouping Preference Rules. Music Perception, vol.4, no.4, 325-360.
- Delin, J. (1986). Review of 'Getting Computers to Talk Like You and Me'. Computational Linguistics, 12, No. 4, 310-312.
- Deutsch, D. & Feroe, J. (1981). The internal representation of pitch sequences in tonal music. Psychological Review, 88, 503-522
- Douglas, S. (1988). Detecting and Repairing Tutoring Failures. Proceedings of the 3rd CERCLE Research Workshop , "Teaching Knowledge and Intelligent Tutoring", Ullswater, April 12-15, Great Britain.
- Draper, S. (1987a). A User-Centred Concept of Explanation. Proceedings of 2nd Alvey

Workshop on Explanation, 1987.

Draper, S. (1987b). Spontaneous Explanation. Proceedings of 2nd Alvey Workshop on Explanation, 1987.

Draper, S. (1987c). What's going on in everyday explanations. To appear in Antaki, C. (Ed.), Analysing lay explanations: a casebook of methods. London, Sage Publications, in press.

Doyle, J. (1979). A Truth Maintenance System. Artificial Intelligence 12, 231-272.

Du Boulay, B., O'Shea, T. & Monk, J. (1981). The black box inside the glass box: presenting computing concepts to novices. Man-Machine Studies 14, 237-249.

Ebcioğlu, K. (1986a). An Expert System for harmonization of Chorales in the Style of J.S. Bach. PhD thesis, Department of Computer Science, University of New York at Buffalo

Ebcioğlu, K. (1986b). An Expert System for Harmonizing Four-Part Chorales. Proceedings of the International Computer-Music Conference, The Hague.

Edwards, D. & Mercer, N. (1987). Common Knowledge: the development of understanding in the classroom. Methuen, London.

Elsom-Cook, M. (1984). Design considerations of an intelligent tutoring system for programming languages. PhD thesis, Warwick University, Great Britain.

Elsom-Cook, M. (1985). Towards a framework for human-computer discourse. In People and Computers: Designing the Interface, (eds.) Johnson, P. & Cook, S., Cambridge University Press. (also as OpenUniversity CITE technical report no. 11).

Elsom-Cook, M. & du Boulay, B. (1986). A PASCAL Program Checker. Proceedings of ECAI '86, vol. II, pp 90 - 95.

Elsom-Cook, M. (1987). Discourse for tutoring. Paper presented at the 3rd International Conference on Artificial Intelligence and Tutoring, University of Pittsburgh LRDC, May 8-10th 1987.

Elsom-Cook, M. & Spensley, F. (1988). Knowledge representation in a tutoring system for procedural skills. CITE Report 36, The Open University, Great Britain.

Elsom-Cook, M. (1988). Introduction to the ECAL System. CITE Report 43, The Open University (GB).

Elsom-Cook, M. (1989a). Dialogue and Teaching Styles. CITE Report No. 62, Open University (GB). To appear in Teaching Knowledge and Intelligent Tutoring, (ed.) Goodyear, P., Ablex Publishing.

Elsom-Cook, M. (1989b). Analysis of a tutorial dialogue. To appear in Elsom-Cook, M. (ed.), Guided-Discovery Tutoring, Paul Chapman Publishing, London.

Elsom-Cook, M. (1989c, in press). Guided-Discovery Tutoring. Paul Chapman Publishing, London.

Elsom-Cook, M. & O'Malley, C. (1989). ECAL: Bridging the gap between CAL and ITS. CITE Report 67, The Open University (GB).

Erman, L.D., Hayes-Roth, F., Lesser, V.R., & Reddy, D.R. (1980). The HEARSAY-II speech-understanding system: integrating knowledge to resolve uncertainty. ACM Surveys, 12 (2), pp. 213-253.

- Farrell, R. (1988). Facilitating Self-Education by Questioning Assumptive Reasoning. Proceedings of the Seventh National Conference on Artificial Intelligence (AAAI-88), pp. 2-6.
- Feigenbaum, E. (1963) The simulation of verbal learning behaviour. Feigenbaum, E. & Feldman, J. (editors) Computers and Thought, McGraw-Hill.
- Fikes, R.E., & Nilsson, N.J. (1971). STRIPS: A new approach to the application of theorem proving to problem solving. Proceedings of the Second International Joint Conference on Artificial Intelligence, Imperial College London, 1st-3rd September 1971.
- Frohlich, D.M. (1988). Conversational Dynamics for Emergent Explanations. Proceedings of 4th Alvey Workshop on Explanation, University of Manchester, September 1988.
- Gilbert, N. (1988). Explanation as Process. Proceedings of 4th Alvey Workshop on Explanation, University of Manchester, September 1988.
- Goguen, J.A., Weiner, J.L. & Linde, C. (1983). Reasoning and Natural Explanation. International Journal of Man-Machine Studies, 19, 521-559.
- Goodyear, P. (1989, in press). Teaching Knowledge and Intelligent Tutoring. (ed.), Ablex Publishing.
- Goldstein, I.P. (1982). The Genetic Graph. Sleeman, D. & Brown, J.S. (eds.), Intelligent Tutoring Systems, Academic Press.
- Greene, J. (1986). Language Understanding: a cognitive approach. Open University Press, Milton Keynes, England.
- Grice, P.H. (1975). Logic and Conversation. Syntax and Semantics, Volume 3: Speech Acts, (eds.) Cole, P. & Morgan, J.L., Academic Press, New York.
- Grosz, B.J. (1977). The representation and use of focus in dialogue understanding. Stanford Research Institute technical note 151. Stanford Research Institute, Menlo Park, California.
- Grosz, B.J. (1979). The representation and use of focus in dialogue understanding. PhD dissertation 1979.
- Grosz, B.J. & Sidner, C.L. (1986). Attention, intentions and the structure of discourse. Computational Linguistics, 12(3), 175-204.
- Hayes-Roth, B, Hayes-Roth, F. Rosenschein, S. & Cammarata, S. (1979). Modelling planning as an incremental opportunistic process. Proceedings of the Sixth International Joint Conference on Artificial Intelligence.
- Heefer, A. & Leman, M. (1986). Chunking as a method of concept acquisition. Proceedings of the European Conference on Artificial Intelligence 1986, volume II, pp. 79-83.
- Hempel, C. (1966). Philosophy of Natural Science. Prentice-Hall, New York.
- Hendrix, G. (1975). Expanding the utility of semantic networks by partitioning. Proceedings of IJCAI 4, pp. 115-121.
- Hinton, G. & Anderson, J.A. (1981). Parallel Models of Associative Memory, Lawrence Erlbaum.
- Jackendoff, R. (1987). Consciousness and the Computational Mind, MIT Press.

- Johnson-Laird, P. (1983). Mental Models. Cambridge University Press.
- Johnson-Laird, P.N. (1986). Reasoning without Logic. In Myers, T., Brown, K. & McGonigle, B. (eds.) Reasoning and Discourse Processes. Academic Press, London.
- Johnson-Laird, P. (1988). The Computer and the Mind. Fontana Press, London.
- Jones, J.A., Scarborough, D.L. & Miller, B.O. (1988). An expert system for harmonic analysis of tonal music. Proceedings of the First Workshop on Artificial Intelligence and Music, St. Paul Minnesota, 1988.
- Kiss, G. (1986). High-Level Dialogue in Man-Machine Interaction. A Survey Commissioned by the Alvey Directorate.
- Laird, J.E., Newell, A. & Rosenbloom, P.R. (1987). SOAR: An Architecture for General Intelligence. Artificial Intelligence 33 (1987) 1-64.
- Lamb, M. & Buckley, V. (1985). A user-interface for teaching piano keyboard techniques. Shackel, B. (ed.) Interact 1985.
- Langlotz, C.P. & Shortcliffe, E.H. (1983). Adapting a consultation system to critique user plans. International Journal of Man-Machine Studies, 19, 479-496.
- Law, A. (1986). The META-HELP Browser: Towards a Practical Interface On-Line Documentation (1). Cognitive Studies Research Papers, CSRP 063, University of Sussex, Brighton, England.
- Lerdahl, F. & Jackendoff, R. (1983). A Generative Theory of Tonal Music. MIT Press.
- Levin, J.A. & Moore, J.A. (1977). Dialogue-Games: Metacommunication Structures for Natural Language Interaction. Cognitive Science 1, Part 4, 395-420.
- Levinson, S.C. (1983). Pragmatics. Cambridge University Press.
- Lischka, C. (1987). Connectionist Models of Musical Thinking. Proceedings of the International Computer Music Conference, University of Illinois at Urbana-Champaign, pp.190-196.
- Longuet-Higgins, C. & Steedman, M. (1970) On Interpreting Bach. Machine Intelligence 6, pp.221-239.
- Longuet-Higgins, C. (1976). Perception of Melodies. Nature vol. 263, pp. 646-653.
- Longuet-Higgins, C. & Lee, C. (1983) The Rhythmic Interpretation of Monophonic Music. In Sundberg, J. (ed.) Studies of Musical Performance, Publications of the Royal Swedish Academy of Music.
- Marsden, A. & Pople, A. (1989). Modelling music cognition as a community of experts. Contemporary Music Review, 3, 1, 29-42.
- McAdams, S. (1987). Music: a science of the mind? Contemporary Music Review, Vol.2 pp. 1 - 61.
- McGuigan, S. & Black, J.B. (1986). Creation and Comprehension of Arguments. In Galambos, J.A., Abelson, R.P. & Black, J.B. Knowledge Structures, Lawrence Erlbaum Associates, Hillsdale, New Jersey.
- Meyer, L.B. (1973). Explaining Music. University of California Press.
- Miller, P.L. (1984). A critiquing approach to expert computer Advice: ATTENDING. London,

- Pitman Publishing.
- Minsky, M. (1977). Frame-system theory. In Johnson-Laird, P.N., & Wason, P.C. (Eds.), Thinking: readings in cognitive science, pp. 355-376, Cambridge University Press.
- Minsky, M. (1981). Music, Mind and Meaning. Computer Music Journal, vol. 5, no. 3 28-44.
- Moorat, J. (1912). Thirty Old-Time Nursery Songs. (arranger). Illustrated by Woodroffe, P. Thames & Hudson.
- Moore, R.C. (1980). Reasoning about knowledge and action. SRI International AI Center Technical Report No. 191.
- Moyse, R. (1989). Knowledge Negotiation Implies Multiple Viewpoints. Artificial Intelligence and Education, (Eds.) Bierman, D., Breuker, J. & Sandberg, J., IOS Publishers, Amsterdam.
- Narmour, E. (1977). Beyond Schenkerism. University of Chicago Press.
- Neisser, U. (1976). Cognition and Reality. W.H. Freeman & Co.
- O'Malley, C.E., Smolensky, P., Bannon, L., Conway, E., Graham, J., Solokov, J. & Monty, M. (1984). A proposal for user centered system documentation. A. Janda (ed.) Human Factors in Computing Systems - I. Amsterdam: North-Holland.
- O'Malley, C. (1987a). Applying studies of natural dialogues to human-computer interaction: are two heads better than one? Proceedings of the 2nd Alvey Workshop on Explanation, pp. 99-108, University of Surrey, Great Britain.
- O'Malley, C. (1987b). Understanding Explanation. Cognitive Science Research Report No. CSRP-88, University of Sussex, May 1987.
- Papert, S. (1980). Mindstorms. The Harvester Press, Brighton, Sussex, Great Britain.
- Pauker, S.G., Gorry, A., Kassirer, J.P., & Schwartz, W.B. (1984). Towards the simulation of clinical cognition: taking a present illness by computer. Clancey, W.J., & Shortcliffe, E.H. (Eds.), Readings in medical artificial intelligence, pp. 131-159, Reading, Massachusetts, Addison-Wesley.
- Peachey, D.R. & McCalla, G.I. (1986). Using Planning Techniques in Intelligent Tutoring Systems. International Journal of Man-Machine Studies, 24, 77-98.
- Peel, J & Slawson, W. (1984). Review of A Generative Theory of Tonal Music, by F. Lerdahl and R. Jackendoff. Journal of Music Theory, 1984 vol.28 part 2, pp. 271-294.
- Petrie-Brown, A. (1989). Intelligent Tutoring Dialogue: The structures of an interaction. Artificial Intelligence and Education, (Eds.) Bierman, D., Breuker, J. & Sandberg, J., IOS Publishers, Amsterdam.
- Poston, E. (1961). The Children's song book. The Bodley Head, London 1961.
- Power, R. (1979). The organisation of purposeful dialogues. Linguistics, 17, 107-152.
- Quillan, M.R. (1968). Semantic Memory. In Minsky, M. (Ed.), Semantic Information Processing, pp. 227-270, Cambridge Massachusetts, MIT Press.
- Reichman, R. (1978). Conversational coherency. Cognitive Science, 3, 283-327.
- Reichman, R. (1985). Getting computers to talk like you and me. Cambridge, Massachusetts,

MIT Press.

- Robinson, J.J. (1982). DIAGRAM: A Grammar for Dialogues. Communications of the ACM, Volume 25, Number 1, 27-47.
- Rosch, E. (1975). Cognitive Representations of Semantic Categories. Journal of Experimental Psychology: General Vol. 104, no.3, pp.192-233.
- Rumelhart, D.E. & McClelland, J.L. (1985) Levels Indeed! A Response to Broadbent. Journal of Experimental Psychology: General 1985, vol.114, no.2, 193-197.
- Rumelhart, D.E. & Norman, D.A. (1985). Representation of Knowledge. Issues in Cognitive Modeling, (eds.) Aitkenhead, A.M. & Slack, J.M., Lawrence Erlbaum.
- Rumelhart, D.E. & Ortony, A. (1977). The Representation of Knowledge in Memory. In Anderson, R.C., Spiro, R.J. & Montague, W.E. (editors) Schooling and the Acquisition of Knowledge, Lawrence Erlbaum.
- Rymazewski, R. (1987). Negotiating Cooperative and Effective Explanations. CERCLE Technical Report No. 39, Dept. of Educational Research, University of Lancaster, GB.
- Sacerdoti, E. (1977). A structure for plans and behavior. Elsevier North-Holland: Amsterdam.
- Schank, R. & Abelson, R.P. (1977). Scripts, Plans, Goals and Understanding. Lawrence Erlbaum, N.J.
- Schegloff, E.A. & Sacks, H. (1973). Opening up Closings. Semiotica, 8 (4), 289-327.
- Schenker, H. (1932). Free Composition. Trans. Oster E. 1979 "Der Frie Satz". Longman.
- Schoenberg, A. (1967). Fundamentals of Musical Composition. Faber & Faber, London.
- Searle, J. (1969). Speech Acts: An Essay in the Philosophy of Language. Cambridge University Press.
- Seely-Brown, J. (1989 in press). Toward a new epistemology for learning. Address given to the Montréal-ITS '88 Conference. To appear in Frasson, C. & Gauthier, J. (eds.) Intelligent Tutoring Systems at the Crossroads of AI and Education. Norwood NJ, Ablex Publishing.
- Self, J. (1988). Use of Belief Systems in Intelligent Tutoring Systems. Unpublished paper, Department of Computing, University of Lancaster, Great Britain. To appear in Proceedings of the 1st European Congress on AI and Training, Lille, France.
- Self, J. (1989). The Case for Formalising Student Models (and Intelligent Tutoring Systems generally). Paper presented the the 4th International Conference on AI & Education, Amsterdam 1989.
- Sidner, C. (1985). Plan parsing for intended response recognition in discourse. Computational Intelligence, 1, 1-10 (1985).
- Simon, H. & Sumner, R. (1968) Pattern in Music. In Formal Representation of Human Judgement (ed.) Kleinmuntz, B., Wiley & Sons.
- Sinclair, J. McH., & Coulthard, R.M. (1975). Towards an Analysis of Discourse: The English Used by Teachers and Pupils. London: Oxford University Press.
- Sleeman, D. & Brown, J.S. (1982). Intelligent Tutoring Systems. Academic Press.
- Sleeman, D. & Hendley, R.J. (1982). ACE: A system which Analyses Complex Explanations. Intelligent Tutoring Systems, (eds.) Sleeman, D. & Brown, J.S., Academic Press.

- Sloboda, J. (1985). The Musical Mind: The Cognitive Psychology of Music. Oxford Psychology Series, Clarendon Press, Oxford.
- Smoliar, S. (1980). A Computer Aid for Schenkerian Analysis. Computer Music Journal, vol. 4, no. 2.
- Steedman, M. (1984). A Generative Grammar for Jazz Chord Sequences. Music Perception, vol.2, no. 1.
- Steele, G. L. (1984). COMMON LISP: The language. Digital Press .
- Steels, L. (1987). The Deepening of Expert Systems. Artificial Intelligence Communications, vol. 0, no.1, August 1987.
- Stevens, A. L. & Collins, A. (1977). The Goal Structure of a Socratic Tutor. BBN Tech. Report No. 3518.
- Stevens, A.L., Collins, A., & Goldin, S. (1978). Diagnosing students' misconceptions in causal models. BBN Technical Report no. 3786.
- Stevens, A., Collins, A. & Goldin, S.E. (1982). Misconceptions in students' understanding. In Intelligent Tutoring Systems, (eds.) Sleeman, D. & Brown, J.S., Academic Press.
- Stoffer, T.H. (1985). Representation of Phrase Structure in the Perception of Music. Music Perception, Winter, vol. 3, no.2, 191-220.
- Stutt, A. (1987). Second Generation Expert Systems. Explanations, Arguments and Archaeology. Human Cognition Research Laboratory Technical Report number 25, The Open University, Milton Keynes.
- Suchman, L.A. (1985). Plans and situated actions: the problem of human-machine communication. XEROX PARC technical report P85-00005, Palo Alto Research Center, Palo Alto.
- Taft-Thomas, M. (1985). VIVACE: A Rule based AI system for composition. Proceedings of the International Computer-Music Conference, Vancouver, pp. 267-274 (Computer Music Association, San Francisco).
- Tenney, J. & Polanski, L. (1980). Temporal Gestalt Perception in Music. Journal of Music Theory, 24,2, pp 205-241.
- Todd, N. (1989). A computational model of rubato. Contemporary Music Review, 3, 1, 69-88.
- Toulmin, S., Rieke, R. & Janik, A. (1979). An Introduction to Reasoning. Macmillan Publishing Co. Inc., New York.
- Walderkine, (1982). From context to text: a psychosemiotic approach to abstract thought. In Beveridge, M. (ed.) Children Thinking Through Language. London: Edward Arnold.
- Weiner, J.L. (1980). BLAH: A system which explains its reasoning. Artificial Intelligence 15, 19-48.
- Wenger, E. (1987). Artificial intelligence and tutoring systems: computational and cognitive approaches to the communication of knowledge. Los Altos, California, Morgan Kaufmann.
- West, R., Cross, I. & Howell, P. (1987). Modelling music perception as input-output and as process. Psychology of Music vol.15, no.1.
- Whyton, W. (1964). 100 Children's Songs. Essex Music, London.

-
- Winkels, R., Breuker, J. & Sandberg, J. (1988). Didactic Discourse in Intelligent Help Systems. Proceedings of ITS-88, Montréal 1988.
- Winograd, T. (1968). Linguistics and the Computer Analysis of Tonal Harmony. Journal of Music Theory, 12 2-49.
- Winograd, T. (1983). Language as a cognitive process. Volume 1: Syntax. Addison-Wesley.
- Winograd, T. & Flores, F. (1986). Understanding Computers and Cognition: A New Foundation for Design. Ablex Publishing Corporation, Norwood N.J.
- Winston, P. (1977). Learning to identify toy block structures. Johnson-Laird, P.N., & Wason, P.C. (Eds.), Thinking : readings in cognitive science, pp. 199-211, Cambridge University Press.
- Woolf, B. & McDonald, D.D. (1983). Human-Computer Discourse in the Design of a PASCAL Tutor. CHI '88 Proceedings, pp. 230-234.
- Woolf, B. (1988). Representing complex knowledge in an intelligent machine tutor. In Self, J. (ed.) Artificial Intelligence and Human Learning, Chapman and Hall Computing, London.

Appendix A: Architecture of an ITS for Musical Structure and Interpretation

This appendix contains a reproduction of a paper presented at the *First Workshop on Artificial Intelligence and Music*, forming part of the AAAI Conference, August 1988, St. Paul, Minnesota, USA. It describes speculative proposals for an Intelligent Tutoring System architecture for extending the work described in this thesis to teaching musical structure and interpretation. The paper includes a description of an earlier version of the dialogue model described in chapters 4 and 5 of this thesis.

An Architecture of an Intelligent Tutoring System for Musical Structure and Interpretation

Michael Baker
Institute of Educational Technology
Centre for Information Technology in Education
The Open University
Milton Keynes MK7 6AA
Great Britain. Telephone: (0908) 653488.
email: MJ_BAKER@UK.AC.OPEN.ACS.VAXA

Paper presented at the "First Workshop on Artificial Intelligence and Music", AAAI Conference, August 1988, St. Paul, Minnesota, USA.

Abstract

The paper describes an architecture for an Intelligent Tutoring System ('INTERPRET') to teach novice musicians to argue critically for their views on structural analysis and interpretation of tonal melodies. A *guided discovery environment* is described which gives guidance on creation of an interpretation, using a computer-based graphic music editing tool. The environment is based on a *tutorial dialogue planning system* and uses teaching strategies which view editing tool actions in terms of a *goal abstraction hierarchy*. A second part of the architecture aims to teach procedural skills in interpretive performance using a gestural input device linked to the system. INTERPRET is being designed and implemented in Common Lisp and flavors on an Apollo Domain workstation.

Introduction

Before an Intelligent Tutoring System ('ITS') is designed and built it should be clear that it fulfills a real educational need. A simple criterion is that the computer should not perform an educational activity any worse than a human teacher, and should not be merely a re-implementation of knowledge which already resides in some other educational medium. In this paper we propose an architecture for an Intelligent Tutoring System ('INTERPRET') for musical structure and interpretation, which is currently being implemented on an Apollo Domain AI workstation, in Common Lisp and Flavors. It aims to give novice instrumentalists and listeners access to educational guidance in these areas of musical activity, without demanding a high degree of technical and theoretical proficiency. Most teachers believe that consideration should be given to these matters from the very outset in music education, but it is a common experience that beginners become so preoccupied with technical aspects such as finger positions, and with the reading of musical notation, that the cognitive and physical load upon them is too great to attend to these finer points.

An orchestral conductor is in the enviable position of being able to control interpretation of a piece, without having to physically play the notes, but few novices have access to such resources and in any case, conducting itself - in 'real time' - is an extremely demanding activity. Computer-based tools exist, however, which can give this kind of power to student and expert musicians. Two principal kinds of tool are *graphic music editors* (Buxton, Reeves, Fedorkow, Smith & Baecker, 1980), and *gestural control devices* (Matthews & Abbott 1980). Such tools do not, however, give novice musicians educational guidance on their use. A current of ITS research since LOGO (Papert 1980) has taken the view that completely 'free discovery' with a computer-based tool leads to a degree of undirectedness which is not only educationally inefficient, but which students often find unrewarding (Elsom-Cook 1984). We shall describe a general architecture for an ITS for musical structure and interpretation which uses these computer-based musical tools as the basis for educational tools called *guided discovery environments*.

In the first of the two main sections of INTERPRET the overall tutorial goal is for the student to create an edited interpretation of a tonal melody, using a graphic editing tool, according to their own preferences, within musical constraints and guidance provided by the system. The music editor is under construction using the Dialog[®] interface design tool. Fundamentally, the architecture of INTERPRET is based on a relationship between musical structure and musical expression, where structure is 'mirrored' in interpretation (Todd 1985; Clarke 1985; Sundberg 1983). Given this relationship, we can use knowledge residing in the system concerning musical structures and their means of possible realisation in an interpretation, to give guidance on the use of such graphic music editing tools. Two main knowledge bases will be used to generate tutorial guidance:

- (i) knowledge of possible musical phrase boundaries;
- (ii) rules for relating structure to means of interpretation.

The first knowledge base is derived using a cognitive model of musical grouping perception (Baker 1988a, in press; Baker 1989, in press) which attempts to model the

kind of knowledge which a listener and teacher might have of the structure of a melody. The model proposes a listener's (or teacher's) knowledge in this domain is often *uncertain* and *incomplete* (the system returns a set of frame-based hypothesised grouping structures, with attendant justifying factors). We would argue that the kinds of tutorial strategies which are required to tutor effectively with such cognitively realistic knowledge will lead to a more educationally effective environment than would be achieved with a relatively shallow and non-humanlike representation of rule-based 'expert' knowledge.

Knowledge of rules for musical performance is derived from existing work in the cognitive psychology of music (Sundberg et al 1983; Todd 1985). We would argue that representing the relationship between musical structure and interpretation in the form of declarative rules alone would not form a suitable basis for Intelligent Tutoring. A more abstract and educationally useful representation of the concepts underlying rules for interpretation is discussed later in the paper. The second main section of the system aims to give guidance in terms of the procedural skills involved in musical performance using a gestural input device (a Roland 'octapad') to enable students to 'perform' their ideal conception in real time.

An outline of the proposed overall tutorial scenario for teaching musical structure and interpretation is as follows:

- (1) the student inputs a tonal melody via mouse selection (dialogue boxes for pitches) and a period of checking that the system has stored what the student wants (quantisation) follows. Alternatively, the student selects a prestored melody (by the student or system).
- (2) (the system analyses structures in the melody)
- (3) a tutorial dialogue is conducted concerning possible musical structures, via the graphic editing tool (both user and system can indicate possible phrase boundaries, and the system can play sections of the melody for demonstration, via an interfaced touch-sensitive Yamaha KX88 synthesiser) and via typed input to a Lisp environment. At some point the user and system reach a measure of agreement on certain phrase boundaries, and the critical dialogue phase ends
- (4) the user is presented with a set of performance variations which can be applied to the graphically represented melody by menu selection (conventional musical notation or 'pianoroll' proportional representation) using a WIMP interface. The interpreted melody can be played via the synthesiser at any point by a simple menu selection. A variety of tutorial strategies are used here to acquaint the student with the relationship between these variations and musical structure, as well as other functions such as vocal modelling. The result is an ideal edit (or more than one version could be saved) which results from explicit cooperation between the student's musical preferences and the musical constraints imposed by the ITS.
- (5) the student 'performs' the preferred performance edit by triggering successive pitches of the melody using a gestural input device (a Roland Octapad), controlling dynamics using the sensitivity of the pad. The system gives educational feedback on the degree of match between the gestural performance and the original edit, suggesting special sections to practice and perhaps suggesting a revision of the original interpretation edit.

(Phases 3, 4 and 5 do not necessarily have to be followed in this order).

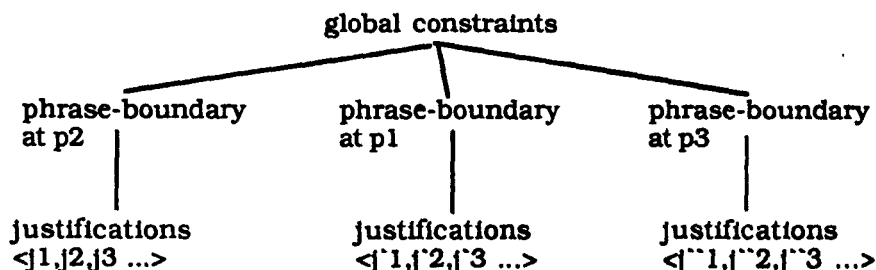
Tutorial dialogues for teaching musical structure

The knowledge-base for musical structures

The model of knowledge of musical structure embodied in INTERPRET was influenced by issues of cognitive validity and of how best to represent the knowledge which a human expert in interpretation of music possesses, in a manner which is most suitable for explicit communication to a student (Wenger 1987). A simple approach to representing knowledge of musical structures would be for a human analyst to prestore this information for a set of pieces in the tutoring system. We have not adopted this approach for a number of reasons. As a pragmatic point, human analysts themselves often disagree over analyses, and at present there are few objective and explicitly agreed principles. We want the system to be able to deal with melodies input and chosen by the student (within limits of the 'musical language' which the system can handle), without having to rely on prestored analyses. Finally, prestored facts alone lack explanatory power: we would like the system to derive structural (or 'grouping') analyses, so that we can use the history of the computation for explaining the occurrence of musical structures.

The method used to derive grouping structures for tonal melodies used in INTERPRET uses frame-based schemata for stereotypical grouping structures to guide a chart parser, using a grammar of chord function (related to those used by Steedman 1984, and which can be abstracted Lerdahl and Jackendoff's 1984 work). The parser is described in Baker 1988a and Baker 1989. It is presently restricted to grouping structures at the phrase-level for very restricted musical genres, but some indication as to further work on sub-phrase levels has been given. For our discussion of tutoring, it is important that the system will not necessarily derive a single correct answer as to phrase structure: a set of hypothesised grouping structures will be derived, together with their attendant justifying factors in terms of harmony, recurring parallelisms (Lerdahl & Jackendoff 1984) and related surface features (the Lerdahl & Jackendoff grouping preference rules). In addition, the system has a set of global constraints on phrase boundary sets, derived from Lerdahl and Jackendoff's well-formedness rules. The knowledge of musical structure available to the ITS can therefore be summarised as in figure 1.

Figure 1 Knowledge of musical structure available to INTERPRET



An example would be

belief 1 - 'phrase boundary at point p1'

j1 - 'exact match in harmony (vi IV ii) in note sets n1 and n2'

j2 - 'exact match in pitch contour, reductional level 3'

belief 2 - 'phrase boundary at p2'

j`1 - 'interval leap of major 6th'

j`2 - 'match at reductional level 2, upward transposition of major 3rd'

Suppose that the student asserted a boundary close to belief 2, which divided the melody into uneven sized groups. A global constraint 'prefer boundaries of even size' could be invoked, and the student would be invited to critically discuss their reasons for claiming that there is a phrase boundary at this point, in opposition to the system's belief set. To simply present the set of hypothesised phrase boundaries and facts which are held to justify belief in them - perhaps in the form of simple text formats - would not form the basis for effective tutorial strategies, since no account is taken of the student's existing knowledge, learning goals and abilities, and ability to integrate this knowledge with new knowledge. The system's knowledge must be communicated in the form of an interactive tutorial dialogue, which achieves tutor and learner goals, is focussed and is relevant.

Planning tutorial dialogues

The approach to tutoring musical structure which we propose is based on generating tutorial dialogues in the form of a *critical argument* (Baker 1988b, in press). We consider that approaches to generating explanations and arguments which are based on generation of single utterance pairs as a result of pruning a record of the computation of rule-based systems (Weiner 1980) or on schematic structures (Stutt 1987) are not suitable for achieving tutorial goals. A suitable alternative is to design a dialogue planning system, building on work on planning in natural language dialogues (Power 1979; Appelt 1982). The planner will operate on sets of system and student possible utterances, grouped into "context spaces" (Reichman 1986), constrained by a set of rules for dialogue focus (Sidner 1986; Grosz 1979).

For a student to be able to take part in a critical discussion in some knowledge domain - in our case, structural analysis of melodies - at least two kinds of knowledge are

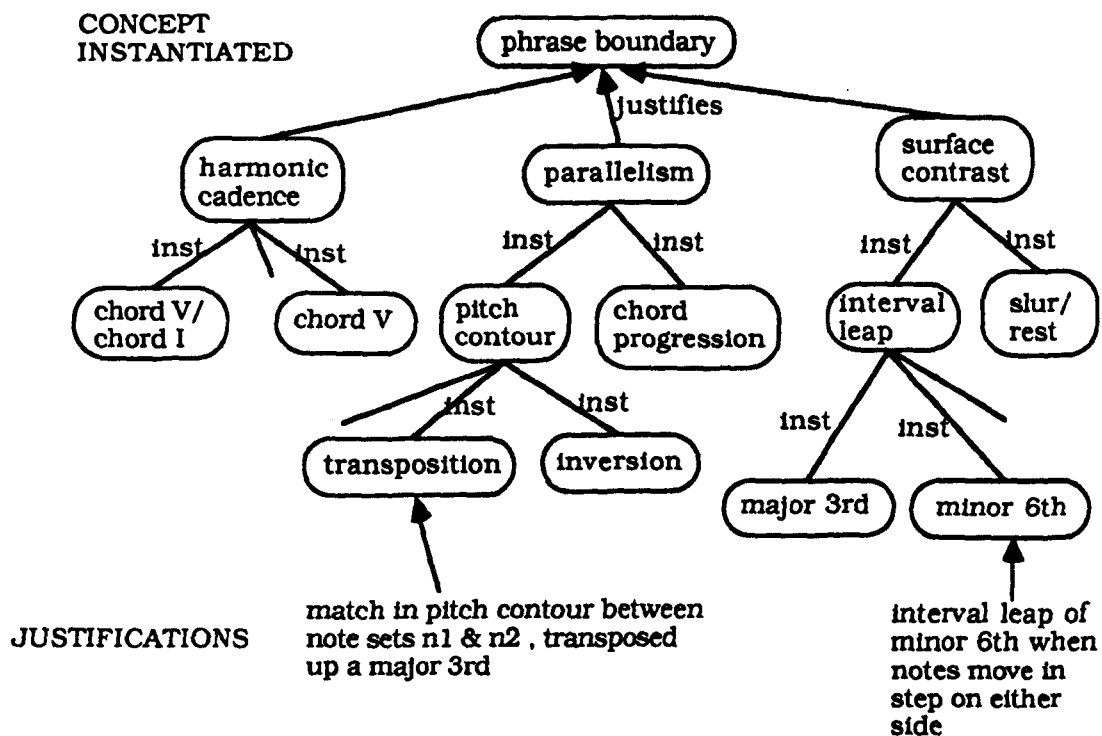
needed:

- (1) knowledge of factors which are relevant to supporting claims, and
- (2) knowledge of how to advance claims and their support in the form of a coherent argument, perhaps in opposition to those put forward by another dialogue participant.

In order to learn the factors which are relevant to supporting claims concerning musical structures a student will need to understand the *concepts* which interrelate different justifying factors, and to be able to integrate these concepts with those which they have already learnt. Of necessity in this knowledge domain, methods of teaching the concept of what a phrase boundary is, for example, by presenting a set of positive, negative and 'near miss' examples (Winston 1977) are precluded - the system has no examples of which it is certain, and must argue for each case. We therefore propose to tutor such concepts by tutoring the concepts which underlie the justifications for positing instances of these concepts. For example, suppose that three justifications for an instance of a phrase boundary are (i) presence of a harmonic cadence, (ii) a change in dynamics, and (iii) a harmonic parallelism. Clearly, the concept of what a cadence is depends on understanding of many others, such as 'chord', 'dominant chord' and many others which may need to be tutored as precursory knowledge. With the second two justifications, what is important is that the second is just one instance of contrast, and that if the student can be taught this, their knowledge will be sufficiently generalised that they will be able to understand that any other form of musical contrast would similarly be a justification here. The same applies to the student learning that there are many different forms of parallelism. These concepts will be represented in the system as a semantic network attached to each type of assertion which the system can make, and which interconnects its set of possible justification types. Figure 2 shows a portion of such a network, which is derived from existing musical theory.

Figure 2

Part of a semantic network relating justifications for possible instances of phrase boundaries:



In an ITS the knowledge representation chosen is usually crucial in influencing a number of features of the system. In this case the semantic network will have the following roles in the system:

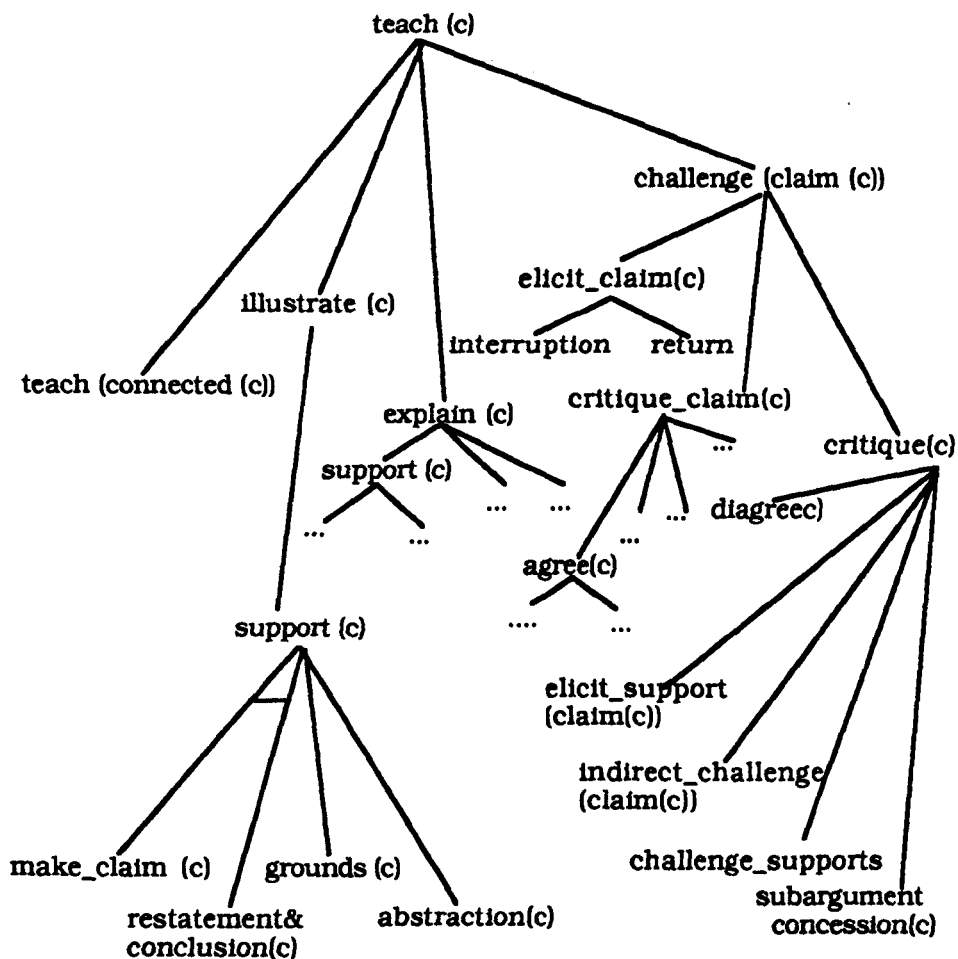
- (i) *choice of tutorial goal* - since the network connects closely related concepts, tutorial goals will be chosen which are connected to previous goals (the goals of teaching that particular concept);
- (ii) *user modelling* - the student's knowledge can be modelled as a simple subset of the tutor's network (we are not addressing the problem of the student possessing or acquiring a quite different conceptualisation of the domain);
- (iii) *dialogue focus* - this relates to choice of tutorial goal, since current dialogue focus will relate to the present position in the network, and dialogue focus shifts will be made smoothly and relevantly to concepts which are implicitly in focus through their connection to the current focus in the network.

In the network shown above, for example, if we are currently teaching the concept 'Interval leap', then the concepts 'Interval' and 'musical surface' are in implicit focus. It would be possible to make the new tutorial goal 'teach interval', with its attendant focus shift. As it stands, such a network could be used in a straightforward fashion (given the reasonably large knowledge engineering effort required to represent it in the first place) to generate explanations for system beliefs by exploiting links in the network, omitting those which we think the student already knows (Weiner 1980) and those which we have not tutored yet, which the student can not integrate with their existing knowledge.

However, this would give little coherent dialogue structure, since the explanation would consist simply of pairs of question/explanation responses. Current research in dialogue and explanation (Draper 1987; O'Malley 1987) has moved towards a view of explanation as an emergent property of an ongoing cooperative dialogue. We are therefore designing a dialogue planning system which will plan top down from current tutorial goals, constrained in a bottom-up fashion by possible relevant focus shifts from the semantic network and possible new topics in terms of the user-model. Global dialogue coherence will be achieved simply in virtue of sets of utterances being part of the same coherent plan, whilst more local coherence is achieved in terms of focus shifts in the network. This view of dialogue focus is closely based on Grosz's seminal work (1979), but in addition, we propose to exploit a large body of research in using semantic networks as models of human memory. Given that a dialogue participant has a current set of explicitly perceived foci, and a set of related topics which may or may not have been explicitly mentioned, our aim is to model this in terms of *spreading activation* (Collins & Quillian 1969; Anderson 1976; McClelland & Rumelhart 1981) from the currently most active foci. Once implicit foci have received a sufficient activation level in the network, then they can be presented to the system's dialogue planner as possible foci to be mentioned and tutored in the subsequent dialogue. As the dialogue progresses, these foci will progressively lose activation (relevance) when not explicitly mentioned.

In its present form, the prototype dialogue planner is simply a 'first person' planner, where the system plans sets of utterances to achieve its tutorial goals. The student's goals and possible utterances are only taken into account via the student model, in terms of possible concepts to teach, and possibly the student's preferred learning style. In further research we hope to be able to make some progress in designing a planner which achieves simultaneous multiple goals of teacher and learner.

Tutorial goals are themselves derived from a realignment of Reichman's (1986) theory of possible conversational moves. They are represented as STRIPS operators (Fikes & Nilsson 1971), with subgoals, predicted effects on the student model and dialogue focus, preconditions and actions which correspond to actual utterances generated by the system (in terms of text generated via dialogue templates - we are not addressing the problem of natural language generation). Figure 3 shows a simplified outline of the hierarchy of planning operators and tutorial goals:

Figure 3 The simplified hierarchy of tutorial planning goals

At the most abstract level, the goal of teaching a concept (in our case, this is the concept 'phrase boundary') can reduce to the subgoals of teaching related concepts (in the case where the student model predicts that the student can not yet apprehend the top level goal of learning the higher level concept and requires precursory knowledge), the making of some claim by the system (illustration and explanation) or the elicitation of some claim from the student which will then be critiqued (Miller 1984). Teaching a related concept simply reinvokes the highest level operator repeatedly until the preconditions of some operator are satisfied which has a non-nil action component. For example, the 'teach(connected(c))' operator may be repeatedly invoked until its preconditions are satisfied, in terms of possible learning goals for the student predicted by the student model. At this point, an explanation of that concept may be given. The difference between an illustration and explanation of a concept is that the former involves giving a concrete (annotated) example, whereas the latter is a more abstract description of the concept. For the concept of phrase boundaries, illustrations are obviously precluded here, but this is not so for its related concepts which can be illustrated with actual musical examples via the interface and synthesiser. Many of the

planning operators are derived from Reichman's (1986) theory of conversational moves, and from a consideration of Levin & Moore's (1977) theory of 'dialogue games'. For example, making an illustration involves making a concrete claim and supporting it. As with the standard components of Reichman's 'support' conversational move, this may be followed by restatement and conclusion of the claim or by further abstract discussion. Once a claim has been elicited from the student (we simply ask something like "are you claiming that there is a phrase boundary at point p?"), the system can follow different kinds of conversational move depending on whether the claim is or is not part of its own belief set. If it shares the student's belief, it will elicit and critique the student's grounds for belief - something which Reichman terms 'indirect challenge' - and similarly with beliefs which it does not share. In this latter case the system can not directly challenge the belief by directly stating that it is false (it does not have the knowledge required to do this). Instead it can employ many of the methods described by Collins and Stevens (1983) in their work on Socratic dialogues, by putting forward a competing hypothesis (in terms of global constraints on boundary sets) or by drawing attention to a competing claim which the student has already made.

The plan interpreter will take an initial top level goal, the student model, dialogue foci record and the knowledge base described earlier as inputs. Using the goal, it generates a plan to achieve it using the operators described. Each operator which has an action part will have certain specified effects on focus (these are specified by Reichman in terms of standard effects of conversational moves) and on the student model. The bottom-up spreading activation model of dialogue focus will constrain which of these effects are possible for coherent dialogue, as will the student model's prediction of which concepts are reasonable learning goals, thus constraining the planner's search through the operator space. The planner will thus return a set of one or more conversational moves (eg 'explain concept harmony'), which may themselves have a number of subparts. For generation of actual text (or musical examples) each action part of a conversational move will have associated local dialogue procedures (Power 1979; Elsom-Cook 1984). Execution of the plan to achieve the tutorial goal will take the form of the system making some utterance (or making a plan to respond to a student utterance in terms of elicited claims) and then the system and student *negotiating* form and content of the next move. Note that we will be avoiding the problem of parsing the student's input by using a constrained menu driven interface for the student's selection of conversational move and its content (filling in slots in frame structures). For example, if the system makes a CLAIM move, then this may be followed by the system making a further claim, the student making a counter claim, the student asking for further explanation and so on. After each conversational move, a phase of negotiation will take place, concerning the next speaker and the new focus, as described by Power (1979) and Levin & Moore (1977). In human dialogue much of this negotiation is implicit, and relies on non-verbal cues. We will adopt the somewhat unsatisfactory approach of explicit negotiation here (Power 1979). Once the move is over and a new speaker and focus chosen (within constraints of the system's perception of relevant focus shifts and suitable learning goals), the plan executor iterates through the planning process to find a new plan, terminating the dialogue when all concepts are taught, believed known to the student, or when the student terminates the dialogue. The overall dialogue will thus consist of many iterations through the space of planning operators, in a manner

analogous to the dialogues generated by Reichman's discourse ATN, alternating with periods of negotiation.

There are a number of unsolved research problems in planning which we would need to address here, including satisfaction of multiple simultaneous goals (of teacher and learner) and planning when effects of actions are unpredictable (abandoning the 'STRIPS assumption', Fikes & Nilsson 1971). We would argue that the student can be taught how to present their knowledge in the form of a critical argument by designing a human computer dialogue system which actually engages them in such a critical argument.

Tutoring interpretation of music

In this section we are concerned with the problem of how best to represent the knowledge which a human expert in interpretation of music possesses in a manner which is most suitable for explicit communication to a student. At first sight, it would appear that the necessary psychological knowledge of the relationship between structure and expression exists for us to be able to incorporate it directly into the knowledge representation of an ITS. We require knowledge of musical structures, and rules which relate structure to interpretation, in the form

<structure f1> , <apply interpretive variation v1, v2, v2 ...>

However, it is now almost a byword of ITS research that the way in which knowledge is represented in rule-based systems is not sufficient alone (witness Clancey's most recent work, Clancey 1988) as a basis for communicating that knowledge to a student. The classic example is of course the early version of GUIDON (Clancey 1987), built on top of the MYCIN expert system for medical diagnosis. Even users who are already experts in the domain find complex chains of rule-based deductions difficult to follow, simply because this does not seem to be an adequate model for human-like problem solving. Later versions of GUIDON seek to address this problem by relating rules into groups and adding further explanatory layers. We may therefore conclude that mere presentation of a set of rules which appear to be in some sense ad hoc and lacking explanation will not be an effective way to facilitate learning of musical interpretation by novices.

Clearly, a deeper level representation of the concepts embodied in these rules is required. Our representation 'pulls' the antecedents and consequents of these performance rules apart to generate a much more abstract representation which we term a *goal abstraction hierarchy*. For example, suppose we have two (simplified) rules

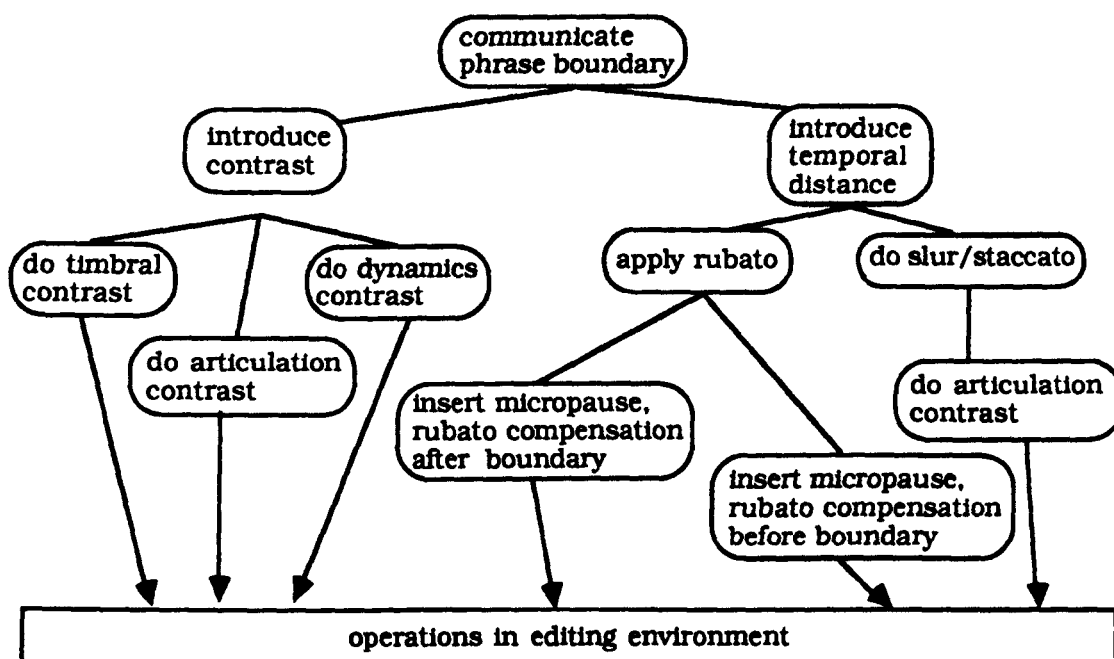
- 1 - <phrase boundary> <do contrast in dynamics>
- 2 - <phrase boundary> <insert micropause>

and that one overall (meta-) goal of interpretation is to communicate structure. We can now view these declarative rules as actual statements of how to perform actions with the graphic editing tool (the right hand sides of the rules) to achieve the common goal (the rule left hand sides) 'communicate phrase boundary'. Given two actions which we want to teach, we need some way of characterising tutorial strategies to present them and tutor the concepts which interrelate alternative actions to achieve goals. We can do this by inserting more abstract goals, which do not actually correspond to actions. For example the goal 'communicate phrase boundary' may have the abstract goal 'introduce

contrast' inserted before editor actions like 'do timbral contrast' and 'do articulation contrast'.

For all performance rules which music psychology has so far formalised (Sundberg et al. 1983; Clarke 1983; Todd 1985) we can produce similar abstractions to generate a complete abstraction hierarchy. Figure 4 shows a small portion of such a network. Given this knowledge representation, and view of rules as actions to achieve goals, a large body of the existing ITS literature can be fruitfully exploited to give us a set of tutorial strategies for recognising students' plans for interaction with the editing tool, and for assisting in their acquisition of these plans.

Figure 4 The goal abstraction hierarchy for performance rules



This problem has been studied mostly in the literature on plans for computing programming (Bonar & Cunningham 1988) and for computer-based procedural skills (Spensley & Elsom-Cook 1988). For example in Bonar and Cunningham's (1988) BRIDGE system for Pascal, the student is led to the production of actual Pascal code by top-down *successive refinement* of an initial top-level goal, through abstract goals such as 'use recursion' or 'use iteration'. Other possible strategies involve either the bottom-up strategy of *cognitive apprenticeship* (Collins & Brown, in press), in which the student watches the system perform the task (an interpretation) and hears the result, and combinations of the two approaches. By using a number of such tutorial strategies, the aim is to allow the student to set each action in its conceptual context, and make the necessary linkages between alternative actions to achieve the same goal. These strategies described so far are highly directive in their constraint on the course of the tutorial interaction, and other less constrained forms of interaction such as discovery learning (Elsom-Cook 1984) could be used where the system was confident that the student had sufficient conceptual understanding to be able to explore the environment

in a directed manner. Developing educationally useful algorithms for choice of most appropriate teaching strategy at any particular point in the interaction is not a trivial problem (a clear algorithm for doing this has been described by Spensley & Elsom-Cook 1988). The overall aim of the system should be to move from an initial position of relative tutorial constraint to one where the student can dispense with tutorial guidance and freely use the environment to achieve their musical goals (the so-called strategy of "decreasing intervention" , Elsom-Cook 1984).

At certain points in the hierarchy, the abstract goals refer to a number of musical concepts (such as rubato and articulation). At this point we can use the completely general dialogue planning architecture described earlier for musical structure to conduct a localised tutorial dialogue (perhaps resulting in an explanation) with the student, in order to extend their understanding of the interpretation editing options available. The relationship between the dialogues on musical structure and those on interpretive variations will not therefore be a matter of a rigid distinction between the two phases (although some discussion of structure will often be a necessary precursor) since it is quite possible that students may wish to revise their views on structure as a result of exploring the effect of various interpretive variations on the way in which the melody is perceived. We have only specified abstraction hierarchies performance rules which have the goal of communicating structure, but similar approaches could be adopted for rules for vocal modelling and musical expression (trivially, the goal 'make the melody more expressive').

At this stage in the design of the system we are not concerned with the precise degree (i.e. mathematically quantified amount) of any particular performance variation which the student may apply. They may introduce any level of dynamic variation as they please, relying on their own musical intuition and their specific acoustic environment. Since the precise degree of a particular interpretive variation required in any musical context is a matter of great importance, it would be possible to build an 'expert' component into the system, for demonstration (and correction) of the appropriate degree of a variation, based on work by Todd (1985) and Friberg & Sundberg (1986).

Tutoring procedural skills in interpretation

So far we have described an ITS architecture which tutors structural analysis and interpretation by consideration of a *static* graphic representation of the music. However, musical interpretation and perception is an activity which is intimately connected with movement in time. As is well documented, (Sloboda 1985) much of what we understand by expression in music is related to bodily gestures, such as the gap inserted between interval leaps being related to necessity for movement on many conventional musical instruments, and the natural tendency to sing louder and faster as melodic pitch rises. For novice performers and listeners, however, the complexity of the physical gestures required by effective performance on an instrument effectively excludes them from active involvement in interpretation. In order to give a degree of gestural involvement in their interpretation, we propose to simplify the gestures required using a gestural input device (a Roland Octapad) where the user's performance gestures are simplified to the triggering of pitches stored in the computer, with precise articulation, rhythm, tempo (fluctuations) and dynamics, by simply striking a pad

with hands or drumstick. The tutorial principle here is analogous to the 'factoring out' of the activities involved in a complex procedural skill, such as flying an aeroplane, commonly used in flight simulators. The user's gestural performance can be captured via MIDI and stored in a file for comparison with their original edited version. By simple analysis of the musically significant differences between the two, educational feedback could be given, with suggestions as to which sections to practise. Again, use of this module of the architecture could be freely mixed with the other two graphic editing phases, since a student may wish to revise their original conception of an interpretation to be more in accordance with their present procedural abilities in gestural performance.

This approach would have a number of advantages over a number of recent computer-based approaches to piano keyboard tuition, which depend on attempting to coerce students into copying specific 'expert' performances (Yokoo & Nagaoka 1985) or correction of variations from the strictly notated text (Steele & Wills 1982). Since the ideal interpretation which the student aims to (gesturally) perform has been derived from a critical discussion with the ITS, the student will understand the rationale behind their own interpretation, and will not be simply copying another performance unthinkingly, or attempting to erase important interpretive variations from their performance in order to play unmusically according to the written notation. This deeper understanding should enable the student's knowledge to be integrated with their existing knowledge, and to transfer to other pieces and styles.

Conclusion

In conclusion, we shall briefly attempt to assess the possible contributions to music education and ITS research which full implementation of this architecture could make. In terms of music education, we would argue that INTERPRET constitutes a valid use of a computer in music education, since it would give novice musicians access to higher level skills which they cannot easily gain by conventional musical means. It is usual in musical practice for students to learn musical analysis and interpretation in a largely intuitive manner, without the kind of explicit discussion of such matters which INTERPRET would engage in. Clearly, there are many more dimensions to interpretive performance than the system can deal with, such as emotional involvement and commitment, and the subtleties of rapport in ensemble playing. We would therefore *not* argue that a system like INTERPRET should serve as a student's sole means of access to such knowledge. Learning such skills by traditional implicit and intuitive methods usually involves decades of musical training. It is possible that discussing these matters in an explicit manner with an ITS like INTERPRET could serve to accelerate learning in the early stages of a novice musician's education. Assessment of such a claim would await empirical evaluation of the fully implemented system.

A significant feature of the system design was development of a cognitive model for perception of musical structures. In an ITS the knowledge representation chosen for the subject domain is usually crucial in determining the form of the tutorial interaction which aims to communicate that domain. Given that our goal is to model a student's perception of musical structure, and to model the structure of dialogues which are coherent and relevant to humans, the development of a representation which is based

on a cognitive model is arguably essential for Intelligent Tutoring. Considerable research effort was expended in developing such a computationally tractable model for music cognition (Baker 1988a; Baker 1989) and we would argue that the development of useful AI based educational tools in music education awaits the further development of such cognitively based knowledge representations.

At present, much of the system is still under detailed theoretical design, but prototype versions exist of the cognitive model, the dialogue planner and the graphic interface. When fully implemented, the system would still be severely restricted in the kinds of musical genres which it could intelligently tutor. Further development in this direction would await further research into computational models of music cognition.

Acknowledgements

Many thanks to my supervisor, Mark Elsom-Cook, for guiding this research, and to Simon Holland for many useful discussions. All the members of the Open University Intelligent Tutoring Systems Research Group have helped me with developing my ideas over the past year or so. Thanks to Fiona Spensley for reading the final draft. The Open University has given me a great deal of support, and I owe a special thank you to Olwyn Wilson for making my life easier. This research is supported by award number 86313101 from the Science and Engineering Research Council of Great Britain.

References

- Anderson, J. (1976). Language, memory and thought. Hillsdale, N.J., Lawrence Erlbaum.
- Appelt, D. E. (1982). Planning Natural Language Utterances to Satisfy Multiple Goals. Technical Note 259, SRI International, California 94025.
- Baker, M.J. (1988a, in press). An Artificial Intelligence Approach to Musical Grouping Analysis. Forthcoming in Contemporary Music Review, special issue on Science and Music.
- Baker, M.J. (1988b, in press). Tutoring with Incomplete and Uncertain Knowledge. Forthcoming in The International Journal of Educational Research, issue on Knowledge Acquisition in Intelligent Learning Systems (c. October 1988).
- Baker, M.J. (1989, in press). A Cognitive Model for Perception of Musical Grouping Structures. Forthcoming in Contemporary Music Review, special issue on Music and the Cognitive Sciences (c. February 1989).
- Bonar, J.G. & Cunningham, R. (1988). BRIDGE: an intelligent tutor for thinking about programming. In Self, J. (ed.) Artificial Intelligence and Human Learning, Chapman Hall, London.
- Buxton, W., Reeves, W., Fedorkow, G., Smith, K.C. & Baecker, R. (1980). A microcomputer-based conducting system Computer Music Journal 4.1, 8-21.
- Clancey, W. (1987). GUIDON: Knowledge based tutoring. MIT Press.
- Clancey, W. (1988). The role of qualitative models in instruction. In Self, J. (ed.) Artificial Intelligence and Human Learning. Chapman & Hall, London.
- Clarke, E. (1985). Structure and Expression in Rhythmic Performance. In Musical Structure and Cognition, eds. Howell, P., Cross, I. & West, R., Academic Press, London.
- Collins, A.M. & Quillian, M.R. (1969). Retrieval time from semantic memory. Journal of verbal learning and verbal behavior, 8, 240-47.
- Collins, A. & Stevens, A.L. (1983). A cognitive theory of inquiry teaching. In Reigeluth, C.M. (ed.) Instructional-design theories and models: an overview of their current status. Lawrence Erlbaum.
- Collins, A. & Brown, J.S. (in press). Cognitive apprenticeship: teaching students the craft of reading, writing and mathematics. In Resnick, L.B. (ed.) Cognition and Instruction: Issues and Agendas. Lawrence Erlbaum.
- Draper, S. (1987). Spontaneous Explanation: report on the dialogue section of the 3rd Alvey workshop on explanation. Proceedings of the 3rd Alvey Workshop on Explanation, University of Surrey (Great Britain), 29th September-1st October, 1987.

- Elsom-Cook, M. (1984). Design Considerations of an Intelligent Tutoring System for Programming Languages. PhD thesis, Warwick University, Great Britain.
- Fikes & Nilsson (1971). STRIPS : A new approach to the application of theorem proving to problem solving. Proceedings of the Second International Joint Conference on Artificial Intelligence, Imperial College, London September 1971.
- Friberg, A. & Sundberg, J. (1986). A Lisp environment for creating and applying rules for musical performance. Proceedings of the International Computer Music Conference, pp.1-3, The Hague, Netherlands.
- Grosz, B. J. (1979). The representation and use of focus in dialogue understanding PhD Dissertation 1979.
- Lerdahl, F. & Jackendoff, R. (1984). A generative theory of tonal music, MIT Press.
- Levin, J.A. & Moore, J.A. (1977). Dialogue-games: metacommunication structures for natural language interaction. Cognitive Science vol. 1, part 4, pp. 395-420.
- Matthews, M.V. & Abbott, C. (1980). The Sequential Drum , Computer Music Journal, Winter 1980 vol 4 no.4.
- McClelland, J.L. & Rumelhart, D.E. (1981). An interactive model of context effects in letter perception: Part I. An account of basic findings. Psychological Review, 1981, 88, 375-407.
- Miller, P. L. (1984) A Critiquing Approach to Expert Computer Advice: ATTENDING, Pitman Publishing, London.
- O'Malley, C. (1987). Applying studies of natural dialogues to human-computer interaction: are two heads better than one? Proceedings of the 2nd Alvey Workshop on Explanation, University of Surrey (Great Britain), January 8-9th, 1987.
- Papert, S. (1980). Mindstorms , Harvester Press, Great Britain.
- Power, R. (1979). The Organisation of Purposeful Dialogues. Linguistics 17, (1979) , 107-152.
- Reichman, R. (1985). Getting Computers to Talk like You and Me. MIT Press, Cambridge Mass.
- Sidner, C.L. (1986). Focussing in the comprehension of Definite Anaphora. In Readings in Natural Language Processing, eds. Grosz, D. , Sparck-Jones, K. & Webber, B.L. , Morgan Kaufman.
- Sloboda, J. (1985). The musical mind: the cognitive psychology of music. Oxford University Press, Oxford, Great Britain.
- Spensley, F. & Elsom-Cook, M. (1988). Dominie: Teaching and assessment strategies Open University CAL Research Group Technical Report No. 74, Milton Keynes, Great Britain.
- Steedman, M. (1984). A generative grammar for jazz chord sequences. Music Perception vol.2, no.1.
- Steele, D. & Wills, B. (1982). Micro-computer assisted instruction for musical performance skills. In Technological Horizons in Education, vol. 9(1), pp 58-60.
- Stevens, A.L. & Collins, A. (1977). The goal structure of a socratic tutor, BBN Technical Report no.3518
- Stutt, A. (1987). Second Generation Expert Systems . Explanations, Arguments and Archaeology. Human Cognition Research Laboratory Technical Report number 25, The Open University, Milton Keynes.
- Sundberg, J. (1983). (ed.) Studies of Musical Performance . Royal Swedish Academy of Music Publications.
- Todd, N. (1985). A Model of Expressive Timing in Music. Music Perception, Fall 1985 vol.3 no. 1, pp. 33-58.
- Weiner, J.L. (1980). BLAH: A system which explains its reasoning. Artificial Intelligence 15,(1980) , 19-48.
- Wenger, E. (1987). Artificial intelligence and tutoring systems: computational and cognitive approaches to the communication of knowledge Morgan Kaufmann Inc., Los Altos, California.
- Winston, P. (1977). Learning to identify toy block structures. In Johnson-Laird, P.N. & Wason, P.C. (eds.) Thinking : readings in cognitive science . Cambridge University Press.
- Yoshinori, Y. & Nagaoka, K. (1985). Computerised methods for evaluating musical performances and for providing instruction techniques for keyboard instruments. Computing and Education, vol.9(2), pp 111-126.

Appendix B: Semantic Network for Musical Structures

(see chapter 5)

```
(setq *checked_network*

' ((PHRASE_BOUNDARY
  ((JUSTIFICATION_OF NIL)
   (HAS_JUSTIFICATION (CONTRAST CHORD_PROGRESSION
                                PARALLELISM))

   (HAS_SUBCONCEPT (PHRASE))
   (SUBCONCEPT_OF NIL)
   (HAS_INSTANCE ())
   (HAS_SUBTYPE NIL)
   (SUBTYPE_OF NIL))
  ((STRENGTH 1) (ACTIVATION_LEVEL 0)
   (EXPLANATORY_TEXT
    ("a phrase boundary is a boundary of a unit of musical structure
     called a phrase. Phrase boundaries are indicated by a complex
     combination of a number of musical factors"))))

(PHRASE
  ((JUSTIFICATION_OF NIL) (HAS_JUSTIFICATION NIL)
   (HAS_SUBCONCEPT NIL)
   (SUBCONCEPT_OF
    (OPENING_PROGRESSION PHRASE_BOUNDARY))
   (HAS_INSTANCE ())
   (HAS_SUBTYPE NIL)
   (SUBTYPE_OF NIL))
  ((STRENGTH 1) (ACTIVATION_LEVEL 0)
   (EXPLANATORY_TEXT
    ("It is difficult to define exactly what a phrase is In
     Fundamentals of Musical Composition Faber 1967 the
     composer Arnold Schoenberg says that The smallest structural
     unit is the phrase a kind of musical molecule consisting of a
     number of integrated musical events possessing a certain
     completeness and well adapted to combination with other
     similar units The term phrase means structurally a unit
     approximating to what one could sing in a single breath"))))

(CONTRAST
  ((JUSTIFICATION_OF (PHRASE_BOUNDARY)) (HAS_JUSTIFICATION NIL)
   (HAS_SUBCONCEPT NIL)
   (SUBCONCEPT_OF NIL)
   (HAS_INSTANCE
    ())
   (HAS_SUBTYPE
    (INTERVAL_LEAP_CONTRAST
     DYNAMICS_CONTRAST
     TIMBRE_CONTRAST
     ARTICULATION_CONTRAST
     MICROPAUSE_CONTRAST))
   (SUBTYPE_OF NIL))
  ((STRENGTH 1) (ACTIVATION_LEVEL 0)
   (EXPLANATORY_TEXT
    ("A contrast is a consistent and perceptible change in some
     musical factor. When we hear a change in some factor the human
     ear tends to unconsciously assume that this change corresponds
     to a structural boundary in the music"))))

(CHORD_PROGRESSION
  ((JUSTIFICATION_OF (PHRASE_BOUNDARY))
```

Appendix B: Semantic Network for Musical Structures

```
(HAS_JUSTIFICATION NIL)
(HAS_SUBCONCEPT (CHORD CHORD_ROOT))
(SUBCONCEPT_OF (HARMONY CHORD_PROGRESSION_PARALLELISM))
(HAS_INSTANCE ())
(HAS_SUBTYPE
(CIRCULAR_PROGRESSION
  OPENING_PROGRESSION
  CLOSING_PROGRESSION
  STRONG_PROGRESSION))
  (SUBTYPE_OF NIL))
((STRENGTH 1) (ACTIVATION_LEVEL 0)
 (EXPLANATORY_TEXT
  ("when we hear a melody the human mind unconsciously infers a
   chord progression which would sound appropriate when played at
   the same time as the melody Special features of this inferred
   chord progression indicate phrase boundaries A chord
   progress"))))

(PARALLELISM
 ((JUSTIFICATION_OF (PHRASE_BOUNDARY))
  (HAS_JUSTIFICATION NIL)
  (HAS_SUBCONCEPT NIL)
  (SUBCONCEPT_OF NIL)
  (HAS_INSTANCE ())
  (HAS_SUBTYPE
   (PITCH_CONTOUR PARALLELISM RHYTHM_PARALLELISM
    CHORD_PROGRESSION_PARALLELISM))
  (SUBTYPE_OF NIL))
 ((STRENGTH 1) (ACTIVATION_LEVEL 0)
  (EXPLANATORY_TEXT
   ("Parallelism refers to any kind of recurring pattern in the
    music so that when a pattern occurs again we tend to hear a
    structural boundary in the music"))))

(CIRCULAR_PROGRESSION
 ((JUSTIFICATION_OF NIL)
  (HAS_JUSTIFICATION NIL)
  (HAS_SUBCONCEPT (PRIMARY_CHORDS SECONDARY_CHORDS))
    (SUBCONCEPT_OF NIL)
    (HAS_INSTANCE ())
    (HAS_SUBTYPE NIL)
    (SUBTYPE_OF (CHORD_PROGRESSION)))
 ((STRENGTH 1) (ACTIVATION_LEVEL 0)
  (EXPLANATORY_TEXT
   ("A circular progression is a kind of chord progression which
    begins on some PRIMARY CHORD progresses to one or more other
    SECONDARY CHORDS and then returns to the original primary chord
    in circular fashion On such a return we tend to hear a
    structural boundary in the music"))))

(PRIMARY_CHORDS
 ((JUSTIFICATION_OF NIL) (HAS_JUSTIFICATION NIL)
  (HAS_SUBCONCEPT (CHORD_I CHORD_V
                     CHORD_IV
                     KEY))
    (SUBCONCEPT_OF
     (SECONDARY_CHORDS CIRCULAR_PROGRESSION
      OPENING_PROGRESSION))
    (HAS_INSTANCE ())
    (HAS_SUBTYPE NIL)
    (SUBTYPE_OF (CHORD)))
 ((STRENGTH 1) (ACTIVATION_LEVEL 0)
  (EXPLANATORY_TEXT
   ("primary chords are those which define the KEY of a piece They
    are chord I chord V and chord IV"))))

(SECONDARY_CHORDS
```


Appendix B: Semantic Network for Musical Structures

```
((JUSTIFICATION_OF NIL) (HAS_JUSTIFICATION NIL)
  (HAS_SUBCONCEPT
    (KEY CHORD_II
      CHORD_III
      CHORD_VI
      CHORD_VII
      PRIMARY_CHORDS))
  (SUBCONCEPT_OF
    (CIRCULAR_PROGRESSION
      OPENING_PROGRESSION))
  (HAS_INSTANCE ())
  (HAS_SUBTYPE NIL)
  (SUBTYPE_OF (CHORD)))
((STRENGTH 1) (ACTIVATION_LEVEL 0)
  (EXPLANATORY_TEXT
    ("secondary chords are all chords in a key other than primary
      chords chord ii chord iii chord vi chord vii"))))

(OPENING_PROGRESSION
  ((JUSTIFICATION_OF NIL) (HAS_JUSTIFICATION NIL)
    (HAS_SUBCONCEPT
      (PRIMARY_CHORDS SECONDARY_CHORDS
        CHORD_V
        CHORD_I
        PHRASE
        KEY))
      (SUBCONCEPT_OF NIL)
      (HAS_INSTANCE ())
      (HAS_SUBTYPE NIL)
      (SUBTYPE_OF (CHORD_PROGRESSION)))
    ((STRENGTH 1) (ACTIVATION_LEVEL 0)
      (EXPLANATORY_TEXT
        ("an opening progression is a kind of CHORD PROGRESSION which
          begins with one or more chords primary or secondary and
          progresses to end on chord V Typically the preceding chords will
          include chord_I if this phrase occurs early in the piece"))))

(CLOSING_PROGRESSION
  ((JUSTIFICATION_OF NIL) (HAS_JUSTIFICATION NIL)
    (HAS_SUBCONCEPT
      (CHORD_V CHORD_I STRONG_PROGRESSION))
      (SUBCONCEPT_OF (KEY))
      (HAS_INSTANCE ())
      (HAS_SUBTYPE NIL)
      (SUBTYPE_OF (CHORD_PROGRESSION)))
    ((STRENGTH 1) (ACTIVATION_LEVEL 0)
      (EXPLANATORY_TEXT
        ("a closing progression is a chord progression which starts with
          any progression of chords but ends with chord V followed by
          chord I. Typically these chords are approached by a STRONG
          PROGRESSION"))))

(CHORD
  ((JUSTIFICATION_OF NIL) (HAS_JUSTIFICATION NIL)
    (HAS_SUBCONCEPT
      (NOTE CHORD_ROOT
        INTERVAL_3RD
        INTERVAL_5TH
        SCALE
        MELODY))
      (SUBCONCEPT_OF
        (CHORD_PROGRESSION CHORD_I
          CHORD_V
          CHORD_IV
          CHORD_I
          CHORD_II
          CHORD_III
```

Appendix B: Semantic Network for Musical Structures

```

                                CHORD_VI
                                CHORD_VII))
(HAS_INSTANCE ())
(HAS_SUBTYPE
  (CHORD_VII CHORD_VI
    CHORD_III
    CHORD_II
    CHORD_I
    CHORD_IV
    CHORD_V
    PRIMARY_CHORDS
    SECONDARY_CHORDS))
(SUBTYPE_OF NIL))
((STRENGTH 1) (ACTIVATION_LEVEL 0)
  (EXPLANATORY_TEXT
    ("A CHORD is simply two or more NOTES which sound simultaneously
      In the kind of music which we are considering - Western Tonal
      Music chords are combinations of three different notes These
      kinds of chords are called TRIADS
    ii a note called the THIRD of the chord which is at an
      INTERVAL of a THIRD above the root
    iii a note called the FIFTH which is at an interval of a
      FIFTH above the root
    The ROOT can be any NOTE which occurs in the SCALE out of which
    the MELODY is formed"))))
(PITCH_CONTOUR_PARALLELISM
  ((JUSTIFICATION_OF NIL) (HAS_JUSTIFICATION NIL)
    (HAS_SUBCONCEPT
      (PITCH_CONTOUR MELODIC_TRANSFORMATION))
    (SUBCONCEPT_OF NIL)
    (HAS_INSTANCE ())
    (HAS_SUBTYPE NIL)
    (SUBTYPE_OF (PARALLELISM)))
  ((STRENGTH 1) (ACTIVATION_LEVEL 0)
    (EXPLANATORY_TEXT
      ("A pitch contour parallelism is a kind of parallelism which
        occurs when the PITCH CONTOUR of one section of a melody matches
        the pitch contour of another The two melodic sections can match
        exactly or they can match when one section is"))))
(PITCH_CONTOUR
  ((JUSTIFICATION_OF NIL) (HAS_JUSTIFICATION NIL)
    (HAS_SUBCONCEPT
      (NOTE MELODY INTERVAL INTERVAL_3RD))
    (SUBCONCEPT_OF
      (MELODY TRANSPOSITION_TRANSFORMATION
        PITCH_CONTOUR_PARALLELISM
        MELODIC_TRANSFORMATION))
    (HAS_INSTANCE ())
    (HAS_SUBTYPE NIL)
    (SUBTYPE_OF NIL))
  ((STRENGTH 1) (ACTIVATION_LEVEL 0)
    (EXPLANATORY_TEXT
      ("A pitch contour is a features of a succession of NOTES forming
        a MELODY which is defined by the pattern of INTERVALS between
        each note. These intervals can be understood in general or
        specific terms for example an interval of a MINOR T"))))
(MELODIC_TRANSFORMATION
  ((JUSTIFICATION_OF NIL) (HAS_JUSTIFICATION NIL)
    (HAS_SUBCONCEPT (PITCH PITCH_CONTOUR))
    (SUBCONCEPT_OF (PITCH_CONTOUR_PARALLELISM))
    (HAS_INSTANCE ())
    (HAS_SUBTYPE (TRANSPOSITION_TRANSFORMATION))
    (SUBTYPE_OF NIL))
  ((STRENGTH 1) (ACTIVATION_LEVEL 0)
```

Appendix B: Semantic Network for Musical Structures

```
(EXPLANATORY_TEXT
  ("A transformation of a melody is a way of changing the actual
    PITCHES of a melody whilst preserving its recognisable MELODIC
    CONTOUR"))))

(PITCH
  ((JUSTIFICATION_OF NIL) (HAS_JUSTIFICATION NIL)
    (HAS_SUBCONCEPT (NOTE))
    (SUBCONCEPT_OF
      (MELODY MAJOR_SCALE
        SEMITONE
        SCALE
        NOTE
        PITCH_REGISTER
        TRANSPOSITION_TRANSFORMATION
        MELODIC_TRANSFORMATION))
    (HAS_INSTANCE ())
    (HAS_SUBTYPE NIL)
    (SUBTYPE_OF NIL))
  ((STRENGTH 1) (ACTIVATION_LEVEL 0)
    (EXPLANATORY_TEXT ("Pitch refers to the highness or lowness of a
      NOTE"))))

(TRANSPOSITION_TRANSFORMATION
  ((JUSTIFICATION_OF NIL) (HAS_JUSTIFICATION NIL)
    (HAS_SUBCONCEPT (PITCH_CONTOUR NOTE PITCH))
    (SUBCONCEPT_OF NIL)
    (HAS_INSTANCE ())
    (HAS_SUBTYPE
      (REAL_TRANSPOSITION TONAL_TRANSPOSITION))
    (SUBTYPE_OF (MELODIC_TRANSFORMATION)))
  ((STRENGTH 1) (ACTIVATION_LEVEL 0)
    (EXPLANATORY_TEXT
      ("Transposition of a PITCH CONTOUR means moving it so that the
        first NOTE begins on another note higher or lower at another
        PITCH"))))

(REAL_TRANSPOSITION
  ((JUSTIFICATION_OF NIL) (HAS_JUSTIFICATION NIL)
    (HAS_SUBCONCEPT (KEY INTERVAL_THIRD))
    (SUBCONCEPT_OF NIL)
    (HAS_INSTANCE ())
    (HAS_SUBTYPE NIL)
    (SUBTYPE_OF (TRANSPOSITION_TRANSFORMATION)))
  ((STRENGTH 1) (ACTIVATION_LEVEL 0)
    (EXPLANATORY_TEXT
      ("Tonal transposition means TRANSPOSITION a PITCH CONTOUR to
        another PITCH where each of the intervals remains exactly the
        same eg a minor third remains a minor third, and so on
        and so the pitch contour may include notes which are not i"))))

(TONAL_TRANSPOSITION
  ((JUSTIFICATION_OF NIL) (HAS_JUSTIFICATION NIL)
    (HAS_SUBCONCEPT (KEY SCALE INTERVAL_THIRD))
    (SUBCONCEPT_OF NIL)
    (HAS_INSTANCE ())
    (HAS_SUBTYPE NIL)
    (SUBTYPE_OF (TRANSPOSITION_TRANSFORMATION)))
  ((STRENGTH 1) (ACTIVATION_LEVEL 0)
    (EXPLANATORY_TEXT
      ("Tonal transposition means TRANSPOSITION a PITCH CONTOUR to
        another PITCH where each of the intervals keeps the same number
        but may change so that the pitch contour stays in the same KEY
        using only the notes of the scale of that key.")))

(RHYTHM_PARALLELISM
  ((JUSTIFICATION_OF NIL) (HAS_JUSTIFICATION NIL)
```

Appendix B: Semantic Network for Musical Structures

```
(HAS_SUBCONCEPT (NOTE MELODY
                  METRIC_VALUE))
(SUBCONCEPT_OF NIL)
(HAS_INSTANCE ())
(INSTANCE_OF NIL)
(HAS_SUBTYPE NIL)
(SUBTYPE_OF (PARALLELISM)))
((STRENGTH 1) (ACTIVATION_LEVEL 0)
 (EXPLANATORY_TEXT
  ("A form of parallelism is that which occurs between the
   METRIC VALUES of NOTES in a MELODY"))))

(METRIC_VALUE
 ((JUSTIFICATION_OF NIL) (HAS_JUSTIFICATION NIL)
  (HAS_SUBCONCEPT (NOTE TEMPO))
  (SUBCONCEPT_OF (RHYTHM_PARALLELISM))
  (HAS_INSTANCE ())
  (HAS_SUBTYPE NIL)
  (SUBTYPE_OF NIL))
 ((STRENGTH 1) (ACTIVATION_LEVEL 0)
  (EXPLANATORY_TEXT
   ("The metric value of a NOTE is the length of a note. Metric
    values are not usually measured in seconds but in terms of
    fractions or proportions of some basic value so that TEMPO is
    measured in terms of numbers of the basic time value of the
    metre "))))

(TEMPO
 ((JUSTIFICATION_OF NIL) (HAS_JUSTIFICATION NIL)
  (HAS_SUBCONCEPT NIL)
  (SUBCONCEPT_OF (RUBATO METRIC_VALUE))
  (HAS_INSTANCE ())
  (HAS_SUBTYPE NIL)
  (SUBTYPE_OF NIL))
 ((STRENGTH 1) (ACTIVATION_LEVEL 0)
  (EXPLANATORY_TEXT
   ("the tempo of a melody is the speed of it Speed is measured in
    terms of unit note values per minute"))))

(CHORD_PROGRESSION_PARALLELISM
 ((JUSTIFICATION_OF NIL) (HAS_JUSTIFICATION NIL)
  (HAS_SUBCONCEPT (CHORD_PROGRESSION HARMONY))
  (SUBCONCEPT_OF NIL)
  (HAS_INSTANCE ())
  (HAS_SUBTYPE NIL)
  (SUBTYPE_OF (PARALLELISM)))
 ((STRENGTH 1) (ACTIVATION_LEVEL 0)
  (EXPLANATORY_TEXT
   ("Patterns can recur in melodies in terms of the HARMONY which
    is unconsciously inferred for sections of the melody"))))

(HARMONY
 ((JUSTIFICATION_OF NIL) (HAS_JUSTIFICATION NIL)
  (HAS_SUBCONCEPT (CHORD_PROGRESSION MELODY))
  (SUBCONCEPT_OF (CHORD_PROGRESSION_PARALLELISM))
  (HAS_INSTANCE ())
  (HAS_SUBTYPE NIL)
  (SUBTYPE_OF NIL))
 ((STRENGTH 1) (ACTIVATION_LEVEL 0)
  (EXPLANATORY_TEXT
   ("Harmony is a chord progression which people hear as sounding
    acceptable to accompany a MELODY"))))

(INTERVAL_LEAP_CONTRAST
 ((JUSTIFICATION_OF NIL) (HAS_JUSTIFICATION NIL)
  (HAS_SUBCONCEPT (MELODY INTERVAL NOTE))
  (SUBCONCEPT_OF NIL)
  (HAS_INSTANCE ()))
```

Appendix B: Semantic Network for Musical Structures

```
(HAS_SUBTYPE NIL)
(SUBTYPE_OF (CONTRAST)))
((STRENGTH 1) (ACTIVATION_LEVEL 0)
(EXPLANATORY_TEXT
  ("A form of contrast in a MELODY is when there an INTERVAL
  occurs between two successive NOTES which is larger than the
  intervals between notes on either side"))))

(DYNAMICS_CONTRAST
  ((JUSTIFICATION_OF NIL) (HAS_JUSTIFICATION NIL)
    (HAS_SUBCONCEPT (DYNAMICS))
    (SUBCONCEPT_OF NIL)
    (HAS_INSTANCE ())
    (HAS_SUBTYPE NIL)
    (SUBTYPE_OF (CONTRAST)))
  ((STRENGTH 1) (ACTIVATION_LEVEL 0)
    (EXPLANATORY_TEXT
      ("A form of contrast in a melody is where dynamics change
      abruptly"))))

(DYNAMICS
  ((JUSTIFICATION_OF NIL) (HAS_JUSTIFICATION NIL)
    (HAS_SUBCONCEPT NIL)
    (SUBCONCEPT_OF (DYNAMICS_CONTRAST))
    (HAS_INSTANCE ())
    (HAS_SUBTYPE NIL)
    (SUBTYPE_OF NIL))
  ((STRENGTH 1) (ACTIVATION_LEVEL 0)
    (EXPLANATORY_TEXT
      ("The dynamics of a melody is its level of loudness or softness
      its volume"))))

(TIMBRE_CONTRAST
  ((JUSTIFICATION_OF NIL) (HAS_JUSTIFICATION NIL)
    (HAS_SUBCONCEPT (PITCH_REGISTER TIMBRE))
    (SUBCONCEPT_OF NIL)
    (HAS_INSTANCE ())
    (HAS_SUBTYPE NIL)
    (SUBTYPE_OF (CONTRAST)))
  ((STRENGTH 1) (ACTIVATION_LEVEL 0)
    (EXPLANATORY_TEXT
      ("A form of contrast occurs when the timbre of the melody
      changes, perhaps by playing in a different pitch register of a
      single musical instrument or by playing on a different kind of
      instrument altogether"))))

(TIMBRE
  ((JUSTIFICATION_OF NIL) (HAS_JUSTIFICATION NIL)
    (HAS_SUBCONCEPT NIL)
    (SUBCONCEPT_OF (TIMBRE_CONTRAST))
    (HAS_INSTANCE ())
    (HAS_SUBTYPE NIL)
    (SUBTYPE_OF NIL))
  ((STRENGTH 1) (ACTIVATION_LEVEL 0)
    (EXPLANATORY_TEXT
      ("Timbre is the specific colour of sound For instance a piano
      has a different sound colour or timbre to a singing voice"))))

(PITCH_REGISTER
  ((JUSTIFICATION_OF NIL) (HAS_JUSTIFICATION NIL)
    (HAS_SUBCONCEPT (PITCH))
    (SUBCONCEPT_OF (TIMBRE_CONTRAST))
    (HAS_INSTANCE ())
    (HAS_SUBTYPE NIL)
    (SUBTYPE_OF NIL))
  ((STRENGTH 1) (ACTIVATION_LEVEL 0)
    (EXPLANATORY_TEXT
```

Appendix B: Semantic Network for Musical Structures

```
("The pitch register simply refers to the general area in which
the pitch of notes in a section of a melody occurs"))))

(ARTICULATION_CONTRAST
  ((JUSTIFICATION_OF NIL) (HAS_JUSTIFICATION NIL)
    (HAS_SUBCONCEPT (ARTICULATION NOTE))
    (SUBCONCEPT_OF (ARTICULATION_MICROPAUSE))
    (HAS_INSTANCE ())
    (HAS_SUBTYPE NIL)
    (SUBTYPE_OF (CONTRAST)))
  ((STRENGTH 1) (ACTIVATION_LEVEL 0)
    (EXPLANATORY_TEXT
      ("A form of contrast occurs when the articulation of notes
      changes"))))

(ARTICULATION
  ((JUSTIFICATION_OF NIL) (HAS_JUSTIFICATION NIL)
    (HAS_SUBCONCEPT (MELODY))
    (SUBCONCEPT_OF (ARTICULATION_CONTRAST))
    (HAS_INSTANCE ())
    (HAS_SUBTYPE (STACCATO LEGATO))
    (SUBTYPE_OF NIL))
  ((STRENGTH 1) (ACTIVATION_LEVEL 0)
    (EXPLANATORY_TEXT
      ("Articulation is simply the property of a melody which refers
      to the general relative lengths of notes and the silences
      between them"))))

(STACCATO
  ((JUSTIFICATION_OF NIL) (HAS_JUSTIFICATION NIL)
    (HAS_SUBCONCEPT (NOTE))
    (SUBCONCEPT_OF NIL)
    (HAS_INSTANCE ())
    (HAS_SUBTYPE NIL)
    (SUBTYPE_OF (ARTICULATION)))
  ((STRENGTH 1) (ACTIVATION_LEVEL 0)
    (EXPLANATORY_TEXT
      ("Staccato articulation simply means that notes are cut very
      short with relatively large silences between successive
      notes"))))

(LEGATO
  ((JUSTIFICATION_OF NIL) (HAS_JUSTIFICATION NIL)
    (HAS_SUBCONCEPT (NOTE))
    (SUBCONCEPT_OF NIL)
    (HAS_INSTANCE ())
    (HAS_SUBTYPE NIL)
    (SUBTYPE_OF (ARTICULATION)))
  ((STRENGTH 1) (ACTIVATION_LEVEL 0)
    (EXPLANATORY_TEXT
      ("Legato articulation means that transitions between notes are
      very smooth with hardly any silence between each successive
      note"))))

(MICROPAUSE_CONTRAST
  ((JUSTIFICATION_OF NIL) (HAS_JUSTIFICATION NIL)
    (HAS_SUBCONCEPT (NOTE))
    (SUBCONCEPT_OF NIL)
    (HAS_INSTANCE ())
    (HAS_SUBTYPE
      (RUBATO MICROPAUSE ARTICULATION_MICROPAUSE))
    (SUBTYPE_OF (CONTRAST)))
  ((STRENGTH 1) (ACTIVATION_LEVEL 0)
    (EXPLANATORY_TEXT
      ("A musical contrast is created when there is a silence between
      two adjacent notes which is perceptibly greater than the
      silence between notes on either side of the two notes"))))
```

Appendix B: Semantic Network for Musical Structures

```
(RUBATO_MICROPAUSE
  ((JUSTIFICATION_OF NIL) (HAS_JUSTIFICATION NIL)
    (HAS_SUBCONCEPT (NOTE RUBATO))
    (SUBCONCEPT_OF NIL)
    (HAS_INSTANCE ())
    (HAS_SUBTYPE NIL)
    (SUBTYPE_OF (MICROPAUSE_CONTRAST)))
  ((STRENGTH 1) (ACTIVATION_LEVEL 0)
    (EXPLANATORY_TEXT
      ("A relatively large silence or rest can be created by starting
        and finishing the first note sooner than usually expected and
        using rubato on the section of music which occurs before the
        silence"))))

(RUBATO
  ((JUSTIFICATION_OF NIL) (HAS_JUSTIFICATION NIL)
    (HAS_SUBCONCEPT (TEMPO))
    (SUBCONCEPT_OF (RUBATO_MICROPAUSE))
    (HAS_INSTANCE ())
    (HAS_SUBTYPE NIL)
    (SUBTYPE_OF NIL))
  ((STRENGTH 1) (ACTIVATION_LEVEL 0)
    (EXPLANATORY_TEXT
      ("Rubato refers to a smooth change in tempo to either slow down
        speed up or usually to do one then the other to return to the
        original tempo"))))

(ARTICULATION_MICROPAUSE
  ((JUSTIFICATION_OF NIL) (HAS_JUSTIFICATION NIL)
    (HAS_SUBCONCEPT (ARTICULATION_CONTRAST))
    (SUBCONCEPT_OF NIL)
    (HAS_INSTANCE ())
    (HAS_SUBTYPE NIL)
    (SUBTYPE_OF (MICROPAUSE_CONTRAST)))
  ((STRENGTH 1) (ACTIVATION_LEVEL 0)
    (EXPLANATORY_TEXT
      ("A relatively large silence or rest can be created by an
        articulation contrast changing from staccato to legato or the
        reverse"))))

(CHORD_V
  ((JUSTIFICATION_OF NIL) (HAS_JUSTIFICATION NIL)
    (HAS_SUBCONCEPT (CHORD KEY))
    (SUBCONCEPT_OF
      (OPENING_PROGRESSION CLOSING_PROGRESSION
        PRIMARY_CHORDS))
    (HAS_INSTANCE ())
    (HAS_SUBTYPE NIL)
    (SUBTYPE_OF (CHORD)))
  ((STRENGTH 1) (ACTIVATION_LEVEL 0)
    (EXPLANATORY_TEXT
      ("Chord V is the triad which has the fifth note of the scale of
        a key as its root"))))

(CHORD_IV
  ((JUSTIFICATION_OF NIL) (HAS_JUSTIFICATION NIL)
    (HAS_SUBCONCEPT (CHORD KEY))
    (SUBCONCEPT_OF (PRIMARY_CHORDS))
    (HAS_INSTANCE ())
    (HAS_SUBTYPE NIL)
    (SUBTYPE_OF (CHORD)))
  ((STRENGTH 1) (ACTIVATION_LEVEL 0)
    (EXPLANATORY_TEXT
      ("Chord IV is the triad which has the fourth note of the scale
        of a key as its root"))))

(CHORD_I
```

Appendix B: Semantic Network for Musical Structures

```
((JUSTIFICATION_OF NIL) (HAS_JUSTIFICATION NIL)
                           (HAS_SUBCONCEPT (CHORD KEY))
                           (SUBCONCEPT_OF
                             (KEY CLOSING_PROGRESSION
                               OPENING_PROGRESSION
                               PRIMARY_CHORDS))
                           (HAS_INSTANCE ())
                           (HAS_SUBTYPE NIL)
                           (SUBTYPE_OF (CHORD)))
((STRENGTH 1) (ACTIVATION_LEVEL 0)
 (EXPLANATORY_TEXT
  ("Chord I is the triad which has the first note of the scale of
   a key as its root"))))

(CHORD_II
 ((JUSTIFICATION_OF NIL) (HAS_JUSTIFICATION NIL)
                           (HAS_SUBCONCEPT (CHORD KEY))
                           (SUBCONCEPT_OF (SECONDARY_CHORDS))
                           (HAS_INSTANCE ())
                           (HAS_SUBTYPE NIL)
                           (SUBTYPE_OF (CHORD)))
 ((STRENGTH 1) (ACTIVATION_LEVEL 0)
 (EXPLANATORY_TEXT
  ("Chord ii is the triad which has the second note of the scale
   of a key as its root"))))

(CHORD_III
 ((JUSTIFICATION_OF NIL) (HAS_JUSTIFICATION NIL)
                           (HAS_SUBCONCEPT (CHORD KEY))
                           (SUBCONCEPT_OF (SECONDARY_CHORDS))
                           (HAS_INSTANCE ())
                           (HAS_SUBTYPE NIL)
                           (SUBTYPE_OF (CHORD)))
 ((STRENGTH 1) (ACTIVATION_LEVEL 0)
 (EXPLANATORY_TEXT
  ("Chord iii is the triad which has the third note of the scale
   of a key as its root"))))

(CHORD_VI
 ((JUSTIFICATION_OF NIL) (HAS_JUSTIFICATION NIL)
                           (HAS_SUBCONCEPT (CHORD KEY))
                           (SUBCONCEPT_OF (SECONDARY_CHORDS))
                           (HAS_INSTANCE ())
                           (HAS_SUBTYPE NIL)
                           (SUBTYPE_OF (CHORD)))
 ((STRENGTH 1) (ACTIVATION_LEVEL 0)
 (EXPLANATORY_TEXT
  ("Chord vi is the triad which has the sixth note of the
   scale of a key as its root"))))

(CHORD_VII
 ((JUSTIFICATION_OF NIL) (HAS_JUSTIFICATION NIL)
                           (HAS_SUBCONCEPT (CHORD KEY))
                           (SUBCONCEPT_OF (SECONDARY_CHORDS))
                           (HAS_INSTANCE ())
                           (HAS_SUBTYPE NIL)
                           (SUBTYPE_OF (CHORD)))
 ((STRENGTH 1) (ACTIVATION_LEVEL 0)
 (EXPLANATORY_TEXT
  ("Chord vii is the triad which has the seventh note of
   the scale of a key as its root"))))

(KEY
 ((JUSTIFICATION_OF NIL) (HAS_JUSTIFICATION NIL)
                           (HAS_SUBCONCEPT
                             (SCALE CHORD_I CLOSING_PROGRESSION))
                           (SUBCONCEPT_OF
```



```

(MAJOR_INTERVAL PERFECT_INTERVAL
    TONAL_TRANSPOSITION
    REAL_TRANSPOSITION
    SECONDARY_CHORDS
    OPENING_PROGRESSION
    PRIMARY_CHORDS
    CHORD_I
    CHORD_II
    CHORD_III
    CHORD_IV
    CHORD_V
    CHORD_VI
    CHORD_VII))
(HAS_INSTANCE ())
(HAS_SUBTYPE (MINOR_KEY MAJOR_KEY))
(SUBTYPE_OF NIL))
((STRENGTH 1) (ACTIVATION_LEVEL 0)
    (EXPLANATORY_TEXT
        ("The key of a piece is based on the scale out of which it is
        constructed The key takes its name from the first note of the scale
        and its type from the type of scale For example a minor scale
        beginning on the note d has the key of d"))))

(STRONG_PROGRESSION
    ((JUSTIFICATION_OF NIL) (HAS_JUSTIFICATION NIL)
        (HAS_SUBCONCEPT
            (CHORD_ROOT INTERVAL
                INTERVAL_5TH
                INTERVAL_4TH))
            (SUBCONCEPT_OF (CLOSING_PROGRESSION))
            (HAS_INSTANCE ())
            (HAS_SUBTYPE NIL)
            (SUBTYPE_OF (CHORD_PROGRESSION)))
    ((STRENGTH 1) (ACTIVATION_LEVEL 0)
        (EXPLANATORY_TEXT
            ("A strong progression is a chord progression where the ROOTS of
            each successive CHORD are an INTERVAL of either a FOURTH or a
            FIFTH apart"))))

(INTERVAL
    ((JUSTIFICATION_OF NIL) (HAS_JUSTIFICATION NIL)
        (HAS_SUBCONCEPT (NOTE SCALE))
        (SUBCONCEPT_OF
            (SCALE SIMPLE_INTERVAL
                INTERVAL_LEAP_CONTRAST
                STRONG_PROGRESSION
                PITCH_CONTOUR))
            (HAS_INSTANCE ())
            (HAS_SUBTYPE
                (INTERVAL_4TH INTERVAL_5TH
                    SIMPLE_INTERVAL
                    COMPOUND_INTERVAL
                    PERFECT_INTERVAL
                    AUGMENTED_INTERVAL
                    DIMINISHED_INTERVAL
                    MAJOR_INTERVAL
                    MINOR_INTERVAL))
            (SUBTYPE_OF NIL))
    ((STRENGTH 1) (ACTIVATION_LEVEL 0)
        (EXPLANATORY_TEXT
            ("An interval simply refers to the distance between two notes
            where those two notes may be consecutive or sounding
            simultaneously. Distance is measured in terms of the inclusive
            number of note names which separate the two notes when counted
            from the lower note "))))

(NOTE
    ((JUSTIFICATION_OF NIL) (HAS_JUSTIFICATION NIL)

```

Appendix B: Semantic Network for Musical Structures

```
(HAS_SUBCONCEPT (INTERVAL_8TH SCALE PITCH))
(SUBCONCEPT_OF
  (TONIC MAJOR_SCALE
    SCALE
    CHORD_ROOT
    INTERVAL
    RUBATO_MICROPAUSE
    MICROPAUSE_CONTRAST
    LEGATO
    STACCATO
    ARTICULATION_CONTRAST
    INTERVAL_LEAP_CONTRAST
    METRIC_VALUE
    RHYTHM_PARALLELISM
    TRANSPOSITION_TRANSFORMATION
    PITCH
    CHORD
    PITCH_CONTOUR))
(HAS_INSTANCE ())
(HAS_SUBTYPE NIL)
(SUBTYPE_OF NIL))
((STRENGTH 1) (ACTIVATION_LEVEL 0)
  (EXPLANATORY_TEXT
    ("A note is an element of a scale. Since scales repeat note
      names after all eight names have been used this means that a
      note can recur with the same name where there is an interval
      of an octave between each recurrence of the same note. ")))

(CHORD_ROOT
  ((JUSTIFICATION_OF NIL) (HAS_JUSTIFICATION NIL)
    (HAS_SUBCONCEPT (NOTE INTERVAL_3RD
      INTERVAL_5TH))
    (SUBCONCEPT_OF
      (CHORD_PROGRESSION CHORD
        STRONG_PROGRESSION))
    (HAS_INSTANCE ())
    (HAS_SUBTYPE NIL)
    (SUBTYPE_OF NIL))
  ((STRENGTH 1) (ACTIVATION_LEVEL 0)
    (EXPLANATORY_TEXT
      ("The root of a chord is the lowest note when the notes are
        rearranged so that the other two are an interval of a third and
        an interval of a fifth above the lowest This will mean that
        the highest note is a third above the note below it"))))

(INTERVAL_3RD
  ((JUSTIFICATION_OF NIL) (HAS_JUSTIFICATION NIL)
    (HAS_SUBCONCEPT NIL)
    (SUBCONCEPT_OF
      (MINOR_INTERVAL MAJOR_INTERVAL
        CHORD
        PITCH_CONTOUR
        CHORD_ROOT))
    (HAS_INSTANCE ())
    (HAS_SUBTYPE NIL)
    (SUBTYPE_OF (SIMPLE_INTERVAL)))
  ((STRENGTH 1) (ACTIVATION_LEVEL 0)
    (EXPLANATORY_TEXT
      ("An interval of a third is the interval between two pitches
        where they encompass three consecutive notes of the scale For
        example the two notes d and f are a third apart because they
        encompass the three notes d e f. Intervals are also a ")))
))
```

Appendix C: Simplified Set of Dialogue Moves used by KANT

(see chapter 5)

```
(setf *dg* #|simplified dialogue goals|#
'({

(DISCUSS
(parameters= (c inst s))
(dialogue_state_preconds=
  (if
    (discussion_over? (ltm_model (dialogue_participant s))
                      (ltm_model s)
                      *dialogue_history*)
    nil
    t))
(negotiation_preconds= ())
(subgoals=
  (OR ((challenge (c inst s))
        (claim (c inst s)))))
(negotiation_effects= ())
(action_effects= nil)
(actions= nil))

(CLAIM
(parameters= (c inst s))
(dialogue_state_preconds=
  (IF
    (not (null c))
    (OR (null *dialogue_history*)
        (AND
          (not (known? c (ltm_model (dialogue_participant s))))
          (exists_ltm_trace? c (ltm_model s))))
    (bind 'c
      (choose_next_concept (ltm_model s)
                           (ltm_model (dialogue_participant s))))))
(negotiation_preconds= ((negotiate s (goal= 'CLAIM)
                                (negotiate s c)))
(subgoals=
  (OR ((concrete_claim (c inst s))
        (abstract_claim (c inst s)))))
(negotiation_effects=
  ((update_dialogue_history '((goal_name= 'CLAIM)
                             (s= s)
                             (c= c)
                             (inst= inst)))))

(action_effects= nil)
(actions= nil))

(CONCRETE_CLAIM
(parameters= (c inst s))
```

Appendix C: Dialogue moves used by KANT

```
(dialogue_state_preconds=
(IF
  (not (null inst))
  (NOT (instance_known?
    inst
    (get_instances c (ltm_model (dialogue_participant s))))))
#|no inst; c must have been bound in parent if it was originally null |#
(bind 'inst
  (choose_next_inst c
    (ltm_model s)
    (ltm_model (dialogue_participant s)))))
(negotiation_preconds= ((negotiate s (goal= 'CONCRETE_CLAIM))
  (negotiate s inst)))
(subgoals=
  (AND-OR ((make_instance_claim (c inst s))
    (support_instance_claim (c (bind inst
      (GET_SUPPORTS inst)) s)) )))
(negotiation_effects=
  ((update_dialogue_history '((goal_name= 'CONCRETE_CLAIM)
    (s= s)
    (inst= inst)))))
(action_effects= nil )
(actions= nil))

(ABSTRACT_CLAIM
(parameters= (c inst s))
(dialogue_state_preconds=
  (OR (NOT (exist_unknown_instances?
    c
    (ltm_model s)
    (ltm_model (dialogue_participant s))))
    (previous_goal? 'concrete_claim c s *dialogue_history*)))
(negotiation_preconds= ((negotiate s (goal= 'ABSTRACT_CLAIM))))
(subgoals=
  (AND-OR ((make_abstract_claim (c inst s))
    (support_abstract_claim ((bind c
      (GET_SUPPORTS C)) inst s)))))
(negotiation_effects= ((update_dialogue_history
  '((goal_name= 'ABSTRACT_CLAIM)
    (c= c)))))
(action_effects= nil)
(actions= nil))

(MAKE_INSTANCE_CLAIM
(parameters= (c inst s))
(dialogue_state_preconds=
  (and (not (null inst))
    (NOT (instance_known?
      inst
      (get_instances c (ltm_model (dialogue_participant s)))))))
(negotiation_preconds=
  ((negotiate s (goal= 'MAKE_INSTANCE_CLAIM))))
(subgoals= nil)
(negotiation_effects=
  ((update_dialogue_history '((goal_name= 'MAKE_INSTANCE_CLAIM)
    (c= c)))))
```

Appendix C: Dialogue moves used by KANT

```
(inst= inst))))

(action_effects=
  ((update_ltm_models '((s= s) (c= c) (inst= inst))))
(actions= (INFORM s 'MAKE_INSTANCE_CLAIM c inst)))

(SUPPORT_INSTANCE_CLAIM
  (parameters= (c inst s))
  (dialogue_state_preconds=
    (and (not (null (cadr inst)))
      (bind 'inst
        (AND
          (NOT (NULL inst))
          (exist_ltm_traces? c inst (ltm_model s))
          (exist_ltm_traces? c inst
            (ltm_model (dialogue_participant s))
            inst))))
    #|inst is here a list of justifications|#
    (negotiation_preconds=
      ((negotiate s (goal= 'SUPPORT_INSTANCE_CLAIM))))
    (subgoals= nil)
    (negotiation_effects=
      ((update_dialogue_history '((goal_name= 'support_instance_claim)
        (s= s)
        (c= c)
        (inst= inst))))))
  (action_effects=
    ((update_ltm_models '((s= s) (c= c) (inst= inst))))
  (actions= (INFORM s 'SUPPORT_INSTANCE_CLAIM c inst)))
  #|inst was bound to list of specific supports|#

(MAKE_ABSTRACT_CLAIM
  (parameters= (c inst s))
  (dialogue_state_preconds=
    (IF
      (equal s 'SYSTEM)
      (and (exists_ltm_trace? c (ltm_model s))
        (null inst)
        (NOT (known? c (ltm_model (dialogue_participant s)))))
      nil))
  (negotiation_preconds=
    ((negotiate s (goal= 'MAKE_ABSTRACT_CLAIM))))
  (subgoals= nil)
  (negotiation_effects=
    ((update_dialogue_history '((goal_name= 'MAKE_ABSTRACT_CLAIM)
      (s= s)
      (c= c)))))
  (action_effects= ((update_ltm_models '((s= s) (c= c)))))
  (actions= (INFORM s 'MAKE_ABSTRACT_CLAIM c inst)))

(SUPPORT_ABSTRACT_CLAIM
  (parameters= (c inst s))
  #|c should come ready bound to list of support type nodes|#
  (dialogue_state_preconds=
    (AND (NOT (NULL c))
      (bind 'C
        (UNKNOWN_ltm_traces
```

Appendix C: Dialogue moves used by KANT

```

      c
      (ltm_model (dialogue_participant s))))))
(negotiation_preconds=
  ((negotiate s (goal= 'SUPPORT_ABSTRACT_CLAIM))
    (negotiate s c)))
(subgoals= nil)
(negotiation_effects=
  ((update_dialogue_history '((goal_name= 'SUPPORT_ABSTRACT_CLAIM)
    (c= c))))))
(action_effects= ((update_ltm_models '((s= s) (c= C))))))
(actions= (INFORM s 'SUPPORT_ABSTRACT_CLAIM c inst)))

(CHALLENGE
(parameters= (c inst s))
(dialogue_state_preconds=
  (AND
    (or (equal 'MAKE_INSTANCE_CLAIM
      (eval (get_parameter_value 'goal_name=
        (last_turn *dialogue_history*))))
      (equal 'SUPPORT_INSTANCE_CLAIM
        (eval (get_parameter_value 'goal_name=
          (last_turn *dialogue_history*))))))
    (exists_ltm_trace? c (ltm_model s))
    (equal c (eval (get_parameter_value 'c=
      (last_turn *dialogue_history*))))
    (bind 'inst (eval (get_parameter_value 'inst=
      (last_turn *dialogue_history*))))))
(negotiation_preconds= ((negotiate s (goal= 'CHALLENGE))
  (negotiate s C) #|renegotiated for challenge|#
  (negotiate s inst)))
(subgoals=
  (OR ((claim_shared (c inst s))
    (claim_not_shared (c inst s)))))
(negotiation_effects=
  ((update_dialogue_history '((goal_name= 'CHALLENGE)
    (c= c)
    (s= s)
    (inst= inst))))))
(action_effects= nil)
(actions= nil))

(CLAIM_SHARED
(parameters= (c inst s))
(dialogue_state_preconds=
  (IF
    (member_inst inst (get_instances c (ltm_model s)))
    inst
    nil))
(negotiation_preconds= ((negotiate s (goal= 'CLAIM_SHARED))))
(subgoals=
  (AND-OR ((agree_supports (c inst s))
    (new_supports (c inst s))
    (disagree_supports (c inst s))
    (complementary_claim (c inst s)))))
(negotiation_effects=
```

Appendix C: Dialogue moves used by KANT

```
((update_dialogue_history '((goal_name= 'CLAIM_SHARED)
                           (inst= inst))))
(action_effects= nil)
(actions= nil))

(CLAIM_NOT_SHARED
 (parameters= (c inst s))
 (dialogue_state_preconds=
  (IF
   (not (member_inst inst (get_instances C (ltm_model s))))
   t
   nil))
 (negotiation_preconds= ((negotiate s (goal= 'CLAIM_NOT_SHARED))))
 (subgoals= ((complementary_claim (C inst s))))
 (negotiation_effects=
  ((update_dialogue_history '((goal_name= 'CLAIM_NOT_SHARED)
                             (inst= inst))))
 (action_effects= nil)
 (actions= nil))

(COMPLEMENTARY_CLAIM
 (parameters= (c inst s))
 (dialogue_state_preconds=
  (and (not (null (cadr inst)))
   (bind 'inst
    (choose_one_inst
     (matching_supports? (get_instances c (ltm_model s))
      inst))))
 (negotiation_preconds=
  ((negotiate s (goal= 'COMPLEMENTARY_CLAIM))))
 (subgoals= ())
 (negotiation_effects=
  ((update_dialogue_history '((goal_name= 'COMPLEMENTARY_CLAIM)
                             (s= s))))
 (action_effects=
  ((update_ltm_models '((s= s) (c= C) (inst= inst))))
 (actions= (INFORM s 'COMPLEMENTARY_CLAIM c inst))

(AGREE_SUPPORTS
 (parameters= (c inst s))
 (dialogue_state_preconds=
  (and (not (null (cadr inst)))
   (bind 'inst
    (list (car inst)
     (get_agreed_supports
      inst
      (member_inst inst (get_instances c (ltm_model s)))))))
 (negotiation_preconds= ((negotiate s (goal= 'AGREE_SUPPORTS))
  (negotiate s C))
 (subgoals= ())
 (negotiation_effects=
  ((update_dialogue_history '((goal_name= 'AGREE_SUPPORTS)
                             (c= C))))
 (action_effects=
  ((update_ltm_models '((s= s) (c= C) (inst= inst))))
 (actions= (INFORM s 'AGREE_SUPPORTS c inst))
```

```

(DISAGREE_SUPPORTS
(parameters= (c inst s))
(dialogue_state_preconds=
(and (not (null (cadr inst)))
(bind 'inst
(list (car inst)
(get_disagreed_supports
inst
(member_inst inst (get_instances c (ltm_model s))) ))))
(negotiation_preconds= ((negotiate s (goal= 'DISAGREE_SUPPORTS))
(negotiate s C)))
(subgoals= ())
(negotiation_effects=
((update_dialogue_history '((goal_name= 'DISAGREE_SUPPORTS)
(c= C)))))
(action_effects=
((update_ltm_models '((s= s) (c= C) (inst= inst)))))
(actions= (INFORM s 'DISAGREE_SUPPORTS c inst)))

(NEW_SUPPORTS
(parameters= (c inst s))
(dialogue_state_preconds=
(and (not (null (cadr inst)))
(bind 'inst
(get_new_supports
inst
(member_inst inst (get_instances c (ltm_model s))) ))))
(negotiation_preconds= ((negotiate s (goal= 'NEW_SUPPORTS))
(negotiate s C)))
(subgoals= ())
(negotiation_effects=
((update_dialogue_history '((goal_name= 'NEW_SUPPORTS)
(c= C)))))
(action_effects=
((update_ltm_models '((s= s) (c= C) (inst= inst)))))
(actions= (INFORM s 'NEW_SUPPORTS c inst)))

```